PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO

OBJECT DETECTION IN VIDEOS USING PRINCIPAL COMPONENT PRUSUIT AND CONVOLUTIONAL NEURAL NETWORKS

Tesis para optar al grado de Magíster en Procesamiento de Señales e Imágenes Digitales que presenta

ENRIQUE DAVID TEJADA GAMERO

Dirigido por

PAUL RODRIGUEZ VALDERRAMA

San Miguel

Marzo 22, 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
ESCUELA DE POSGRADO
MAESTRÍA EN PROCESAMIENTO DE SEÑALES E IMÁGENES DIGITALES

Object Detection in videos using Principal Component Pursuit and Convolutional Neural Networks

In partial fulfillment of the requirements for the Degree of

Master in Digital Signal and Image Processing

in the Graduate School of the Pontificia Universidad Católica del Perú.

Submitted by

Enrique David Tejada Gamero

Thesis supervised by:

Paul Rodríguez

Examining commitee members:

Paul Rodríguez

Daniel Racoceanu

César Carranza

San Miguel

March 22, 2018

To my parents and brothers, for their support, love and strength provided.

Acknowledgments

”Es ist nicht genug zu wissen, man muß auch anwenden; es ist nicht genug zu wollen, man muß auch tun.“ - Johann Wolfgang von Goethe

I deeply thank my parents, for giving me the values and education that allowed me to become who I am today. To my mother, for his constant support and strength that kept me going on. To my father, for his invaluable advices that help me throughout my life and studies. To my brother Eduardo, for setting an example to follow.

I also want to thank all the professors that guided me during my graduate studies, especially my advisor Paul Rodriguez for enabling me to complete this thesis. Thanks to the Fondecyt grant 169-2015-FONDECYT-DE. Thanks to the members of the comittee for their helpful comments. Thanks to all my friends from the different labs at PUCP, for their advice and friendship.

Abstract

Object recognition in videos is one of the main challenges in computer vision. Several methods have been proposed to achieve this task, such as background subtraction, temporal differencing, optical flow, particle filtering among others. Since the introduction of Convolutonal Neural Networks (CNN) for object detection in the Imagenet Large Scale Visual Recognition Competition (ILSVRC), its use for image detection and classification has increased, becoming the state-of-the-art for such task, being Faster R-CNN the preferred model in the latest ILSVRC challenges. Moreover, the Faster R-CNN model, with minimum modifications, has been succesfully used to detect and classify objects (either static or dynamic) in video sequences; in such setup, the frames of the video are input "as is" i.e. without any pre-processing.

In this thesis work we propose to use Robust PCA (RPCA, a.k.a. Principal Component Pursuit, PCP), as a video background modeling pre-processing step, before using the Faster R-CNN model, in order to improve the overall performance of detection and classification of, specifically, the moving objects. We hypothesize that such pre-processing step, which segments the moving objects from the background, would reduce the amount of regions to be analyzed in a given frame and thus (i) improve the classification time and (ii) reduce the error in classification for the dynamic objects present in the video. In particular, we use a fully incremental RPCA / PCP algorithm that is suitable for real-time or on-line processing.

Furthermore, we present extensive computational results that were carried out in three different platforms: A high-end server with a Tesla K40m GPU, a desktop with a Tesla K10m GPU and the embedded system Jetson TK1. Our classification results attain competitive or superior performance in terms of F-measure, achieving an improvement ranging from 3.7% to 97.2%, with a mean improvement of 22% when the sparse image was used to detect and classify the object with the neural network, while at the same time, reducing the classification time in all architectures by a factor raging between 2% and 25%.

Keywords

Object detection, convolutional neural networks, Principal Component Pursuit

Contents

Chapter 1

Introduction

Object recognition in videos is one of the main challenges in computer vision and of high importance for video surveillance, human activity, vehicle counting and others [1]. Several challenges have been proposed (Pascal [2], COCO [3], ILSVRC [4]) in order to find the best classifier. Through time several methods and algorithms were proposed, the most remarkable being those that used features extractors, such as SURF [5], BRISK [6], HOG [7] and SIFT [8], to obtain characteristics of an image and classify them with a Support Vector Machine (SVM). These models provided an acceptable performance with TOP-5 errors (i.e. the fraction of test images for which the correct label is not among the five labels considered most probable by the model), as low as 0.26172 by 2012 [4]. However in 2012 [9] introduced a new model using Deep Convolutional Neural Networks (CNN) (the model proposed can be seen in Figure (1)) which represented a turning point in image detection and classification. Since that moment Deep Learning (DL) [10], as shown in Figure 2, has increasingly influenced the field of object recognition, and, nowadays most classification techniques involve a CNN model [11, 12, 13], achieving better results in terms of mean Average Precision (mAP) [2], F-measure and error rate .

In order to correctly classify objects over the whole image, it is necessary to segment the regions to be classified. Most of the work related to object detection and classification attempt to solve the problem by analyzing all the image and then determining all the objects that are present, some of them based on

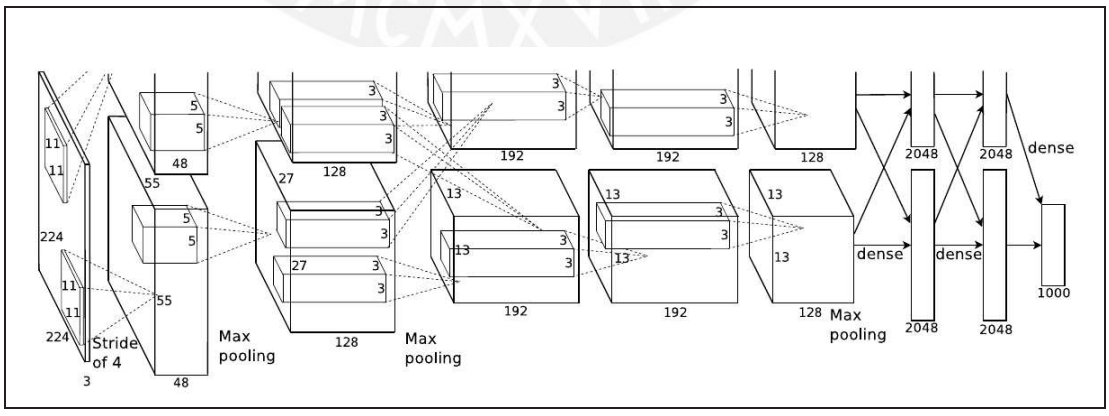


Figure 1: Structure of the Alexnet CNN proposed in [9]

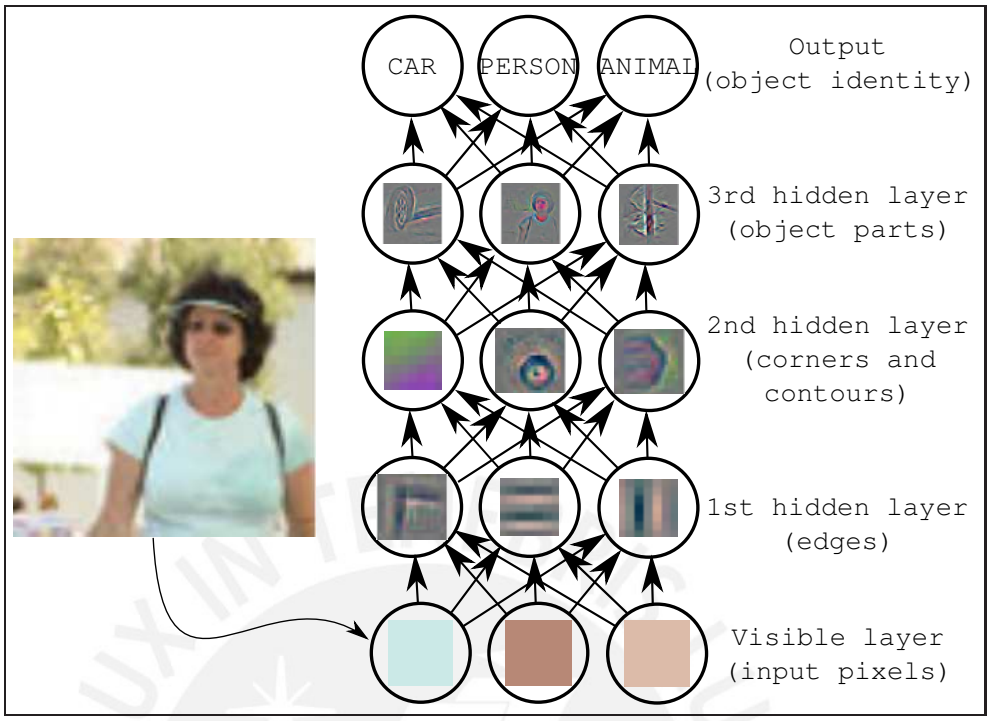


Figure 2: Illustration of a deep learning model. Image from the Deep Learning book [14].

grouping super-pixels, such as Selective Search [15], Contrained Parametric Min-Cuts (CPMC) [16], and others based on sliding windows such as [17] and [18]. One of the methods that gives high rate of overlap with respect to the groundtruth is the Selective Search (SS) method [15]. From a general point of view, the SS method is based on a hierarchical grouping: first initial regions are created, and then via a greedy algorithm such regions are grouped based on similarities. This process continues until the whole image is a single region, generating the possible object locations with high box overlapping, achieving a high recall of MABO[1] for their "Fast" and "Quality" methods respectively in the Pascal 2007 test dataset [2]; however, it must be noted that this method has a high detection time that makes it unsuitable for real-time processing. This information gathered with the SS method can be used in different types of classifiers. For instance, [11] made use of the selective search method to find regions which were then classified by a Convolutional Neural Network. Recently, a Fully Convolutional Network (FCN) for Semantic Segmentation was introduced in [12]. By adapting the fully connected layer of well known classifiers, such as LeNet [19], AlexNet [9] among others, into convolutions that covers their entire input regions they achieved a semantic segmentation of the image, with state-of-the-art segmentation results for the PASCAL VOC dataset. In [10], the Faster R-CNN model was introduced, which is a modification of the model used in [11], specifically two changes were made to the model: (i) the SS method was changed by a Region Proposal Network (RPN) model, which is based on the FCN model, to generate region proposals and (ii) in order to reduce the computational cost, the features from the convolutional layers of the CNN were shared with the RPN. The proposed regions returned by the RPN are used in a ROI Pooling Layer along with the feature maps to classify the objects. With this approach the Faster R-CNN model achieved state-of-the-art measures in detection and classification as

[1]Mean Average Box Overlapping

can be noted in [10]. This model achieved a low classification time, averaging 200ms for 300 proposed regions (the authors used an NVidia GPU K40 @ 875 Mhz and an Intel Xeon CPU E5-2650 v2 @ 2.60GHz). Moreover, several recent works related to object detection in images and videos, are based on Faster-RCNN model, such as DeepID-Net [13] and the solution proposed by the NUIST team in the ILSVRC challenge of 2016 [20].

1.1 Description of the proposed method

Although the classification performance of images with CNN achieve state-of-the-art in terms of F-measure and accuracy, the amount of memory used to classify such images makes it restrictive for mobile applications [21]. In this regard, a new approach is necessary to solve the classification problem of moving objects, that can the classification time with a classification performance similar or superior to the state-of-the-art classifiers and could provide a mean to further reduce the memory footprint. Several pre-processing techniques have been previously proposed in order to improve the CNN's performance in image/object classification The most common techniques include mean image subtraction [22], whitening [23]. Moreover, [24] proposed a method based on dimensionality reduction by applying Principal Component Analysis (PCA) on the image prior to the classification task. In order to improve the overall performance of detection and classification of, specifically, the moving objects in a video sequence, we propose to use a video background modeling pre-processing step. Video background modeling is a ubiquitous pre-processing step in several computer vision applications, used to detect moving objects in digital videos. There are several models for this task, e.g. based on the computation of the median [25] or histograms [26], support vector machines [27], subspace learning [28], neural networks [29, 30]. More recent models are based in PCP [31, 32] and Outlier Pursuit [33] among other variants. To the best of our knowledge, no pre-processing techniques have been previously reported for the case where the objective is to classify the moving objects in video sequences. This motived us to apply a suitable RPCA/PCP algorithm to perform a video background modeling pre-processing step and cascade it with the Faster R-CNN. We hypothesize that such pre-processing step, which segments the moving objects from the background, would reduce the amount of regions to be analyzed in a given frame and thus (i) improve the classification time, and (ii) reduce the error in classification for the dynamic objects present in the video. In particular, we use a fully incremental RPCA / PCP algorithm [34, 35] that is suitable for real-time or on-line processing.

Chapter 2

Methodology

In this chapter we summarize the methods that will be used for the proposed model. This chapter is organized in two sections. Section 2.1 describes the Video Background Modeling pre-processing step, specifically Principal Component Pursuit in its incremental form, followed by the ghost suppresion (gs-incPCP) and the ℓ_1-ball projection (ℓ_1B-PCP) variations. In Section 2.2, we give a description of Convolutional Neural Networks and their influence in image classification, as well as a detailed description of the Faster R-CNN model used for the classification step.

2.1 Video Background Modeling via Principal Component Pursuit

In this section we give a brief overview of the RPCA / PCP method, with a particular focus on the incremental PCP algorithm [34, 35] (which in turn is based on [36]), which is entangled with the Faster R-CNN in order to improve the overall classification performance.

Video background modeling is a ubiquitous pre-processing step in several computer vision applications, used to detect moving objects in digital videos. There are several models for this task, e.g. based on the computation of the median [25] or histograms [26], support vector machines [27], subspace learning [28], neural networks [29, 30]. More recent models are based in PCP [31, 32] and Outlier Pursuit [33] among other variants. To the best of our knowledge, recursive projected compressive sensing (ReProCS) [37, 38] along with Grassmannian robust adaptive subspace tracking algorithm (GRASTA) [39], ℓ_p-norm robust online subspace tracking (pROST) [40], Grassmannian online subspace updates with structured sparsity (GOSUS) [41] and the incremental PCP (incPCP) [35] are the only PCP-like methods for the video background modeling problem that are considered to be incremental. However, except for incPCP, these methods have a batch initialization/training stage as the default/recommended initial background estimate. GRASTA and GOSUS can perform the initial background estimation in a non-batch fashion, however the resulting performance is not as good as when the default batch procedure is used; see [35, Section 6]. pROST is closely related to GRASTA, and it shares the same restrictions. All variants of ReProCS also use a batch initialization stage.

2.1.1 Incremental Principal component Pursuit

In particular, PCP was introduced in [31] as the non-convex optimization problem given by (1)

$$\arg\min_{L,S} \ \text{rank}(L) + \lambda\|S\|_0 \ \text{ s.t. } D = L + S\,, \tag{1}$$

where $D \in \mathbb{R}^{m\times n}$ is the observed video of n frames, each of size $m = N_r \times N_c \times N_d$ (rows, columns and depth or channels respectively), $L \in \mathbb{R}^{m\times n}$ is a low rank matrix representing the background and $S \in \mathbb{R}^{m\times n}$ is a sparse matrix representing the foreground (moving objects).

While most PCP algorithms, including the Augmented Lagrange Multiplier (ALM) and inexact ALM (iALM) algorithms [42, 43] are directly based on the convex relaxation (2)

$$\arg\min_{L,S} \|L\|_* + \lambda\|S\|_1 \ \text{ s.t. } D = L + S, \tag{2}$$

this is not the only possible tractable problem that can be derived from (1). As it is shown in [36, 35] (3) is also proper convex relaxation of (1)

$$\arg\min_{L,S} \frac{1}{2}\|L+S-D\|_F^2 + \lambda\|S\|_1 \quad \text{s.t. } \text{rank}(L) \le r\,. \tag{3}$$

Furthermore, in [36] the numerical solution for (3) was carried out via the alternating optimization

$$L_k^{(j+1)} = \arg\min_{L} \|L_k + S_k^{(j)} - D_k\|_F^2 \ \text{s.t. } \text{rank}(L) \le r \tag{4}$$

$$S_k^{(j+1)} = \arg\min_{S} \|L_k^{(j+1)} + S_k - D_k\|_F^2 + \lambda\|S_k\|_1, \tag{5}$$

where sub-problem (4) can be solved by computing a partial (with r components) SVD of $D - S^{(j)}$. In (4) $L_k = [L_{k-1}\ \mathbf{l}_k]$, $S_k = [S_{k-1}\ \mathbf{s}_k]$ and $D_k = [D_{k-1}\ \mathbf{d}_k]$, where $\mathbf{d}_k$ is the next frame available from the input video, L_k and S_k are the low-rank and sparse representations respectively. In the first, i.e. $j = 0$, we can solve (4) via

$$L_k^{(1)} = partialSVD(D_k - S_k^{(0)}) \tag{6}$$

Since $D_k - S_k^{(0)} = [D_{k-1} - S_{k-1}\ \mathbf{d}_k] = [L_{k-1}\ \mathbf{d}_k]$, and $L_{k-1} = U_r\sigma_r V_r^T$, (6) can be solved via the incremental thin SVD procedure. The solution of (5) is found with a shrinkage applied to the current estimate

$$s_k^{(1)} = shrink(\mathbf{d}_k - \mathbf{l}_k^{(1)}, \lambda) \tag{7}$$

where

$$shrink(x,\varepsilon) = sign(x)max\{0, |x| - \varepsilon\}\,. \tag{8}$$

and $\mathbf{l}_k^{(1)}$ is the las column of the current estimate $L_k^{(1)}$. In the next inner loop $(j = 1)$ for solving (3) we will have

$$\begin{aligned} L_k^{(2)} &= partialSVD(D_k - S_k^{(1)}) \\ &= partialSVD([D_{k-1} - S_{k-1}\mathbf{d}_k - \mathbf{s}_k^{(1)}]), \end{aligned}$$

which can be computed using the thin SVD replace procedure. A full detail of the Incremental and rank-1 modifications for thin SVD can be found on [35].

2.1.2 Ghosting Supression for Incremental PCP

Ghosting is a phenomenon that occurs when an element from the background is assigned to the foreground, or when actual moving objects produce phantoms or smear replicas. In the context of PCP, this occurs when a moving object suddenly stops, a stationary object suddenly starts moving or a moving object occludes a high contrast background object. The effect of these phenomena will be noticeable if a binary mask is computed from the sparse component. To overcome this problem a variant of the incremental PCP algorithm was proposed in [44]. Given two low-rank components $\mathbf{l}_k^{(n_1)}$ and $\mathbf{l}_k^{(n_2)}$, with $n_1 \ll n_2$ will be different if a video event's interpretation differs over a given time frame, e.g., but not limited to, when a moving object suddenly stops. This differences will be shown in the sparse components, $\mathbf{s}_k^{n_1}$ and $\mathbf{s}_k^{n_2}$, as ghosts. If a binary mask is computed from the each sparse component, i.e. $\mathbf{m}_k^{n_1}$ and $\mathbf{m}_k^{n_2}$, these mask will show the moving objects and the ghosts. As shown in [35], the intersection of these masks will provide only the moving objects, and the union of the masks complement,i.e. $\mathcal{B}_k = \sim m_k^{n_1} \cup \sim m_k^{n_2}$, will include the pixels of the background that are not occluded by a moving object. With $\mathcal{B}_k$ we can (i) generate an adaptative λ for each frame k instead of the globally fixed λ from (5) and (ii) generate a "new" input frame $\hat{\mathbf{d}}^{(n)} = \mathbf{d}_k \odot \mathcal{B}_k + \mathbf{l}_k^n \odot (1 - \mathcal{B}_k)$, where $\odot$ is element-wise multiplication (Hadamard product), which will be used to replace the effect of the previously processed frame $\mathbf{d}_k$ with the use of the downdate modification for the thin SVD. The complete implementation of this can be found in Algorithm 1 of [35].

2.1.3 Incremental PCP via projections onto the ℓ_1-ball

Although a theoretical guidance for selecting an optimal regularization parameter λ is given in [32], tipically this parameter is chosen heuristically. Recently a novel convex relaxation of (1) was proposed in [45]

$$\arg\min_{L,S} \frac{1}{2}\|L+S-D\|_F^2 \quad \text{s.t. } \|S\|_1 \leq \tau,\ \text{rank}(L) \leq r\,, \tag{9}$$

and, as with other incremental PCP algorithms, (9) can be solved in an incremental fashion, and the parameter τ can be adaptively estimated for every frame. The same approach used in Section 2.1.2 can be applied to (9) and solve it via an alternating optimization

$$L_k^{(j+1)} = \arg\min_{L} \|L_k + S_k^{(j)} - D_k\|_F^2 \text{ s.t. } \text{rank}(L_k) \leq r \tag{10}$$

$$S_k^{(j+1)} = \arg\min_{S} \|L_k^{(j+1)} + S_k - D_k\|_F^2 \text{ s.t. } \|S_k\|_1 \leq \tau, \tag{11}$$

where $L_k = [L_{k-1}\ l_k]$, $S_k = [S_{k-1}\ s_k]$ and $D_k = [D_{k-1}\ d_k]$. The minimization of (10) can be computed via the incremental thin SVD procedure, while the minimizer of (11) is the projection of $(d_k - l_k)$ onto the ℓ_1-ball

$$s_k^{(1)} = \underset{\ell_1-ball,\tau}{\text{prox}}\ (\mathbf{d}_k - \mathbf{l}_k^{(1)}) \quad \text{s.t. } \|S\|_1 \leq \tau\,, \tag{12}$$

where

$$\underset{\ell_1-ball,\tau}{\text{prox}}\ (\mathbf{u}) = \min_x \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2 \quad \text{s.t. } \|x\|_1 \leq \tau\,, \tag{13}$$

is the projection onto the ℓ_1-ball. If $\|\mathbf{u}\|_1 \leq \tau$, then $\mathbf{x}^* = \mathbf{u}$ is the solution to (13); however this is rarely observed in practices, and thus we asume $\|\mathbf{u}\|_1 > \tau$. Then the optimal solution $\|\mathbf{x}^*\|_1 = \tau$, and the solution to (13) is given by shrinkage

$$x^* = shrink(\mathbf{u}, \lambda(\tau)) \tag{14}$$

where $\lambda(\tau)$ is a threshold that depends of τ and is usually found by sorting the elements of $\mathbf{u}$ in decreasing order. While there are several algorithms to solve (13), the one proposed in [46] is used since it can be easily parallelize in several architectures, including CUDA.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) were introduced in 1989 in [47]. The main difference from classical neural networks, is the presence of a **Convolutional Layer** in which three steps are performed:

1. A convolutional operation between the input data and a learned kernel that produces a linear activation, which in turn
2. The ouput of convolution is applied to a non linear activation function, such as the ReLU (Rectified Linear Unit) and finally
3. A pooling operation is performed in the data obtained from the ReLU. There are several pooling functions, being the **max pooling** [48] one of the most used in Convolutional Neural Networks.

A sample model of a CNN can be appreciated in Figure 3 where all the stages mentioned above are depicted.

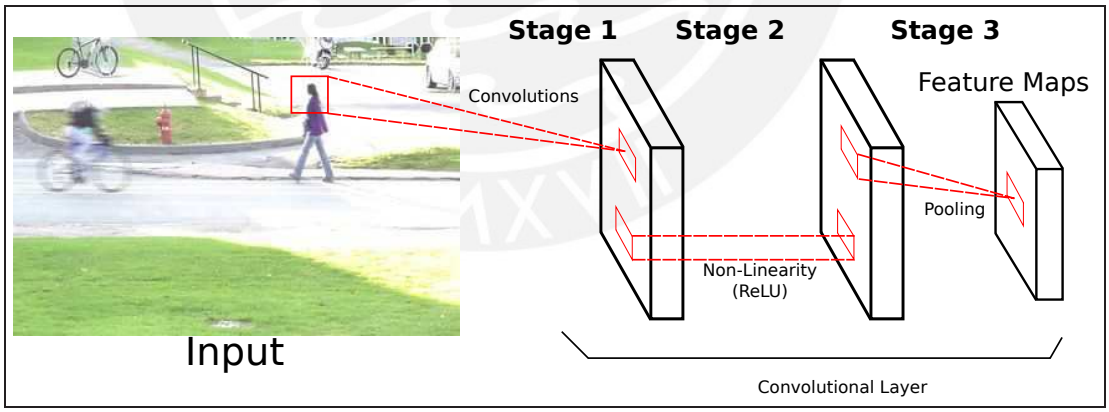


Figure 3: Example of the stages of a Convolutional Layer

CNN's were conceived from the work performed in [49, 50, 51]. In these studies, the authors analized the mammalian visual system behavior and determined that some neurons responded more strongly to certain type of patterns, such as oriented bars. These neurons belong to the V1 cortex, also known as primary visual cortex. Here, features are detected hierarchically, i.e., first some coarse features are detected and more complex features are built based on them. In this sense, CNNs can be compared

in behavior to the V1 cortex, as the first layer detects basic features and subsequent convolutional layers detect more complex ones. Nowadays, the state-of-the-art for image classification is achieved by Convolutional Neural Networks (CNN), as it is shown with the top methods presented in the Imagenet Large Scale Visual Recognition Challenge (ILSCVR) [4]. In 2012 this approach gained more attention with the work of [9] where they achieved an outstanding test error rate of 15.4% while the next best entry in that challenge achieved an error of 26.2% error. Recently, a new model was presented, Faster R-CNN [10], based on the Fast R-CNN model presented in [52] and proposing a Region Proposal Network (RPN) for generating the region proposals, instead of the Region of Interest (RoI) pooling layer of [52]. This particular method will be detailed in following Sections.

2.2.1 Fully Convolutional Neural Network (FCN)

A Fully Convolutional Network (FCN) is a type of neural network in which all the layers perform convolutional operations. One of the first works to adapt a Convolutional Neural Network (CNN) into a FCN was presented in [53], where they extended the CNN described in [47] to recognize a complete set of digits, instead of an isolated digit. It is due to the nature of the Layers of an FCN that it posess some advantages over traditional CNN. One of these advantages is that an FCN can be trained end-to-end, and thus, learn features in all the layers. On the other hand, as all the layers of an FCN are convolutional, the FCN computes a non-linear filter, instead of a non-linear function compared to regural CNNs.

FCN are used in several applications, such as image restoration [54] and Semantic Segmentation [12]. An example of transformation from a CNN into a FCN to perform Semantic Segmentation can be noted in Figure 4 where the fully conected layers were changed by convolutional layers. The model presented in [12] is of special interest since it provides an state-of-the-art segmentation using an FCN, as can be seen in Figure 5, which will base for the RPN proposed by [10].

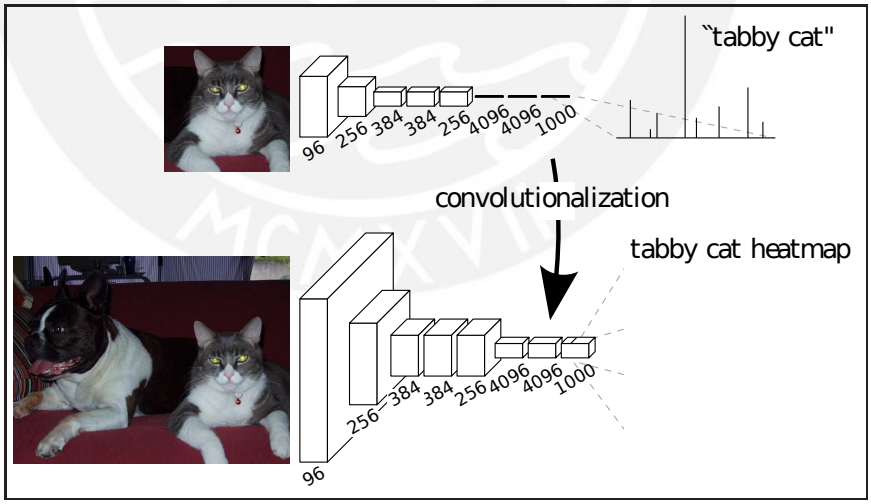


Figure 4: Example of transformation from a CNN to a FCN by changing the fully conected layers into convolution layers. Image taken from [12].

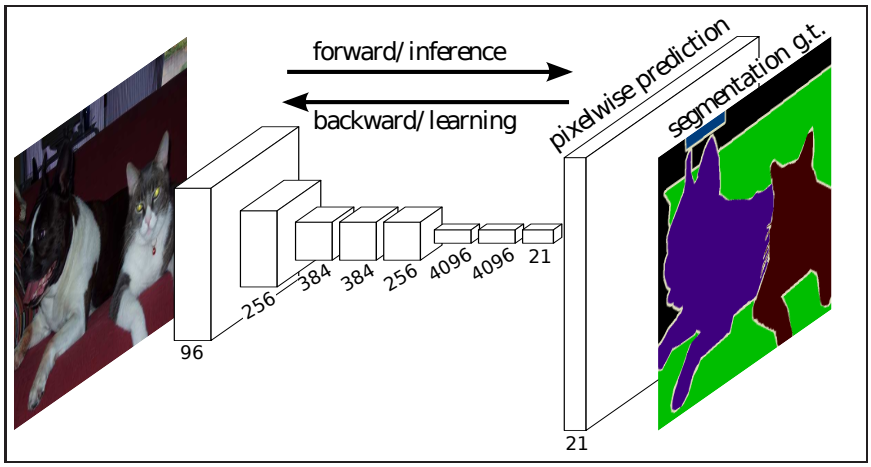


Figure 5: Example of deep prediction using a semantic segmentation performed by an FCN. Image taken from [12].

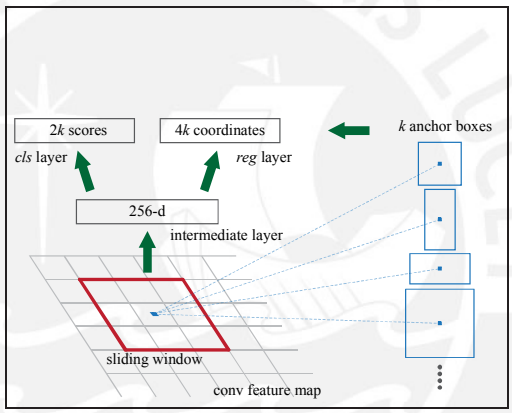


Figure 6: The RPN model generates k anchors and returns $2k$ scores and $4k$ coordinates. Image taken from [10].

2.2.2 Region Proposal Network (RPN)

The Region Proposal Network (RPN) was first described in [10] and is based on the work of [12]. This model is a specific type of Fully Convolutional Network which shares a common set of convolutional layers with the an object detection network. An example of this can be seen in Figure 7. This network takes as input an $n \times n$ window of the input convolutional feature map. Each sliding window will generate anchors, to determine the location of the region as welll as a probability estimated of an object. These anchors have 3 different shapes and the test is performed at 3 different scales yielding a total of 9 anchors at each sliding position. Each window will provide k possible regions, the *reg* layer of the RPN will provide the bounding boxes coordinates of the regions and the *cls* layer will provide an estimate probability of object. The model proposed is shown in Figure 6.

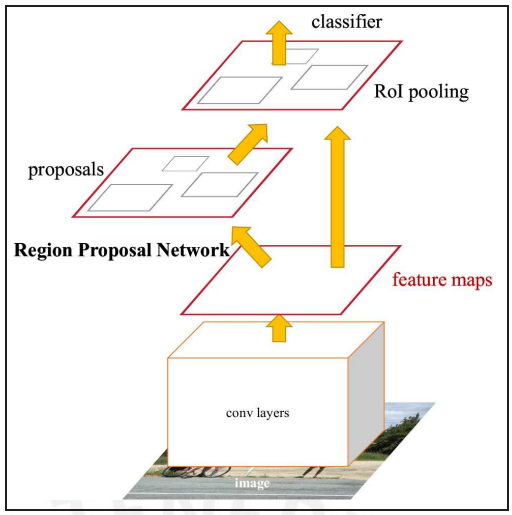


Figure 7: Model proposed in [10]. The Region Proposal Network shares some convolutional layers with the CNN.

2.2.3 Faster R-CNN

The Faster R-CNN model proposed by [10] consists of a deep fully convolutional network that proposes regions, while the second module is a detector based on the Fast R-CNN [52] that base its decision on the proposed regions of the RPN. This model uses the ZF model [55], which has 5 convolutional layers that can be shared with the RPN and the VGG-16[56] wich has 13 convolutional layers available for sharing. Sharing the feature maps between the RPN and the convolutional layers allows a reduction of computational cost and processing time (These results can be found on Table 5 of [10]) , and an increment in the classification performance, achieving state-of the art results for object detection and classification. Figure 7 shows the unified network of the Faster R-CNN.

The Faster R-CNN model has shown great performance in object classification and it has been used as a basis for new models and techniques ([13]) in the different categories of the ILSCVR challenge, obtaining state-of-the-art results for detection and classification. Most models focuse in detection and classification of all the objects in an image, and in the case of videos, this will increase the computational cost. To solve this problem, we propose the use of PCP as a pre-processing step to perform a segmentation of the moving objects in videos and reduce the computational cost and classification time, since less regions are to be found.

Chapter 3

Proposed Method: Sparse pre-processing for Convolutional Neural Networks

For the classification of moving objects in videos, the incremental PCP with ghosting suppression algorithm [44] as well as the incremental PCP via projections onto the ℓ_1-ball were applied. Assuming that for any frame k, the low-rank (**l**) and sparse (**s**) components satisfy

$$\mathbf{d}_k \approx \mathbf{l}_k + \mathbf{s}_k, \tag{1}$$

then a binary mask $\mathbf{m}_k$ was automatically computed via an automatic unimodal segmentation [57], since the absolute value of the sparse representation has an unimodal histogram, from $\mathbf{s}_k$. Then, such mask was applied to the original frame, i.e.

$$\mathbf{u}_k = \mathbf{m}_k \odot \mathbf{d}_k, \tag{2}$$

where $\odot$ represents element-wise product. This step can be observed in Figure 8. The images $\mathbf{u}_k$ were fed to a pre-trained CNN, specifically, the Faster-RCNN [10] model with the "fast" version of ZF net [55] that has 5 convolutional layers and 3 fully-connected layers. This scheme can be seen in Figure 10. Although Faster R-CNN can use other models such as VGG-16 [56] for the classification, the ZF model was chosen due to the hardware restrictions of "Mobile" platform (see Table I). The neural network returns the bounding boxes of the images detected along with the score of classification for each bounding box, and the time needed to classify the objects in the image. This information is used along with the groundtruth for each video to determine the F-measure

$$F = \frac{2 \cdot P \cdot R}{P+R}, \;\; P = \frac{TP}{TP+FN}, \;\; R = \frac{TP}{TP+FP} \tag{3}$$

where P and R stand for precision and recall respectively, and TP, FN and FP are the number of true positive, false negative and false positive pixels, respectively.

Computational Experiments

In order to assess the time performance of the proposed method[1], we have run our experiments in three different hardware platforms, labeled as "Server", "Desktop" and "Mobile". While their particular

[1] To use PCP as a video background modeling pre-processing step, before using the Faster R-CNN model

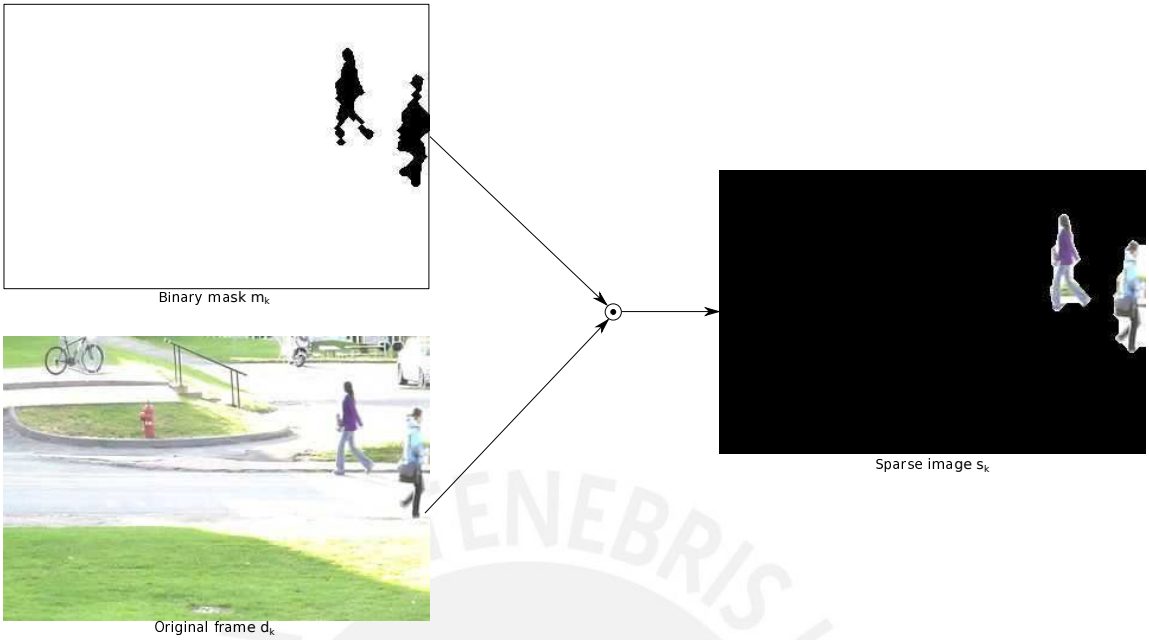


Figure 8: Example of the proposed pre-processing step. The sparse image s_k is obtained by applying the binary mask m_k to the original frame d_k.

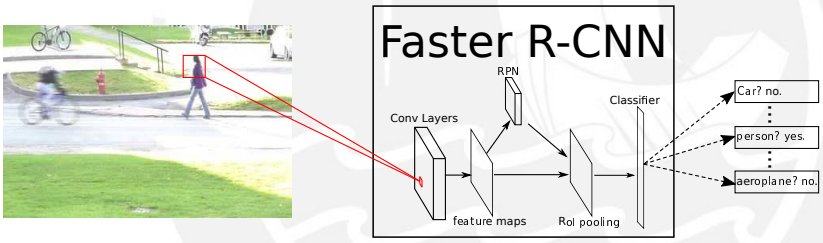


Figure 9: Original Faster R-CNN model. The input image of the CNN is the current frame $\mathbf{d}_k$ from the input video.

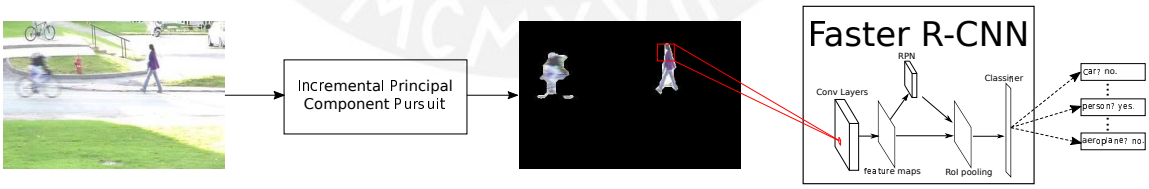


Figure 10: Proposed method with the incremental PCP algorithm as a pre-processing step. The input image $\mathbf{u}_k$ of the CNN is obtained with (2).

characteristics are listed in Table I, we highlight that the main objective of using these different platforms was to factor out any hardware dependency in our experiments.

Platform	CPU	Memory	GPU
Server	32x Intel® Xeon™ E5-2640 v2 @ 2.0GHz 20MB cache	128 GB	2x NVidia Tesla K40m 12GB
Desktop	8x Intel® Core™ i7-2600K @ 3.40GHz 7MB cache	32 GB	2x NVidia Tesla K10m 8GB
Mobile	ARM® Cortex™ A15 @ 2.3GHz 32 KB L1 cache /512KB L2 cache	2 GB	Tegra K1

Table I: Hardware used in the experiments. These platform where chosen as they are considered to be representative architectures for different applications, from high-end Processing (Tesla K40m) to mobile applications (Tegra K1).

The CDNet2014 [58] dataset was selected for the tests since it comprise of several videos with particular characteristics that allow tests of moving object detection in different scenarios. We selected seven from four different categories of the CDNet dataset, some frame samples can be found in Figure 11

- **badWeather**
  - skating: This is a video of people skating in a park in a snowy day. The influence of the snow cause a low contrast between the objects and the background, this is reflected in the low classification accuracy.
- **baseline**
  - highway: This simple video of cars circulating in a highway has no many alterations in the background. The leaves of the trees generates a little of ghosting in the Sparse component of the PCP algorithm.
  - pedestrians: In this video we can observe people walking in a park. The illumination allows good contrast between the objects and the background.
  - PETS2006: This is a benchmark data used to detect abandoned luggage. The high contrast and steady background allows a good sparse segmentation.
- **shadow**
  - backdoor: This video show people walking in an alley with influence of shadows from different objects.
  - busStation: In this video, we can observe people coming out of a bus station. The shadows from nearby buildings and from the people affect the computation of the sparse component.
  - cubicle: This video show people walking inside an office. The shadows cast from objects and the people walking by has the same influence as in the other videos of this category.

(a) Frame 1434 from the skating video

(b) Frame 1435 from the skating video

(c) Frame 797 from the highway video

(d) Frame 798 from the highway video

(e) Frame 570 from the pedestrians video

(f) Frame 571 from the pedestrians video

(g) Frame 115 from the PETS2006 video

(h) Frame 116 from the PETS2006 video

(i) Frame 1851 from the backdoor video

(j) Frame 1852 from the backdoor video

(k) Frame 1019 from the busStation video

(l) Frame 1020 from the busStation video

(m) Frame 2828 from the cubicle video

(n) Frame 2829 from the cubicle video

Figure 11: Sample frames of the videos used from the CDNet2014 dataset

Chapter 4

Results

Three different tests were run in each platform, first the classification was performed on the original images of the videos, the second classification was performed on the segmented images $\mathbf{u}_k$ using the $\mathbf{m}_k$ from the gs-incPCP algorithm, and a third classification was run over the segmented images $\mathbf{u}_k$ obtained with the $\mathbf{m}_k$ from the ℓ_1B-PCP algorithm. The F-measure was calculated for each one of the videos and for each test. In order to compute the F-measure, first we calculate the overlap ratio between the groundtruth bounding boxes and the bounding boxes provided by the Classifier using the Intersection over Union (IoU) method, this ratio allowed us to determine the metrics needed in the F-measure calculation.

	All platforms		
Dataset	**Original frame**: $\mathbf{d}_k$	**Masked frame gs-incPCP**: $\mathbf{u}_k$ (see (2))	**Masked frame ℓ_1B-PCP**: $\mathbf{u}_k$ (see (2))
backdoor	0.7755	**0.8309**	0.8282
busStation	0.1927	**0.3801**	0.3635
cubicle	**0.7505**	0.6008	0.6563
highway	**0.8383**	0.8002	0.8150
pedestrians	0.6094	**0.8842**	0.8780
PETS2006	0.5068	**0.6231**	0.6185
skating	0.4690	**0.4863**	0.3630

Table II: The F-measure computed for the 7 datasets. Results are shown for classification over original frames ($\mathbf{d}_k$) and for masked frames ($\mathbf{u}_k$) (see (2))

The performance given by the F-measure are shown in Table II. We first mention that, unsurprisingly, the performance results are the same for all platform. We can note that for most of the videos, the performance of the F-measure was higher for both pre-processing algorithms, and gs-incPCP achieved a slightly better performance of the F-measure. In Figure 12 through 15 we can observe some classification examples comparing the standard classification and the method proposed with the two PCP algorithms. Although the performance of the proposed method is better for most of the considered test videos, the "cubicle" and "highway" videos are for which the standard classification gave better performance. We can note also that for these cases the ℓ_1B-PCP showed a better performance than the gs-incPCP. The average classification time for each video is shown in Table IV the impact on the time reduction observed when classifying the sparse images over the original images will depend on the

application. It is worth to mention that PCP time depends solely on the image size and not the content. These times were reported in [59] and are reproduced in Table III. As can be seen, the overhead time of the PCP algorithm doesn't substantially affect the time of the proposed mehtod.

Figure 12: Classification Sample of frame 1099 of the video **busStation**. The score for the detected objects improve, and through all the video the classification will provide a better F-measure by only classifying the moving objects.

Figure 13: Classification Sample of frame 595 of the video **pedestrians**. In this example, some objects are misclassified, reducing the F-measure performance.

Dataset	Frame size	PCP average time Desktop GPU	PCP average time Jetson TK1
backdoor	320x240	6.0	16.0
busStation	360x240	6.0	16.0
cubicle	352x240	6.0	16.0
highway	320x240	6.0	16.0
pedestrians	360x240	6.0	16.0
PETS2006	720x576	24.0	54.0
skating	540x360	21.0	49.0

Table III: Average PCP procesing times for each video, all times are in milliseconds. This times are taken from Table 4 in [59].

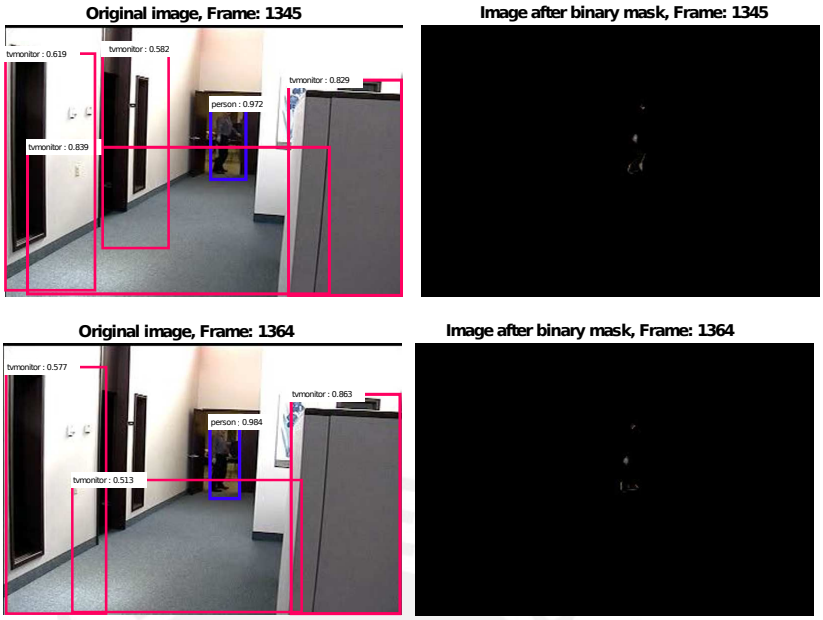


Figure 14: Classification Sample of frames 1345 and 1364 of the video **cubicle**. It can be noted that due to the person standing still for a period of time the PCP Algorithm set him as part of the background. This is a recurrent error where the person stand still for a period of time.

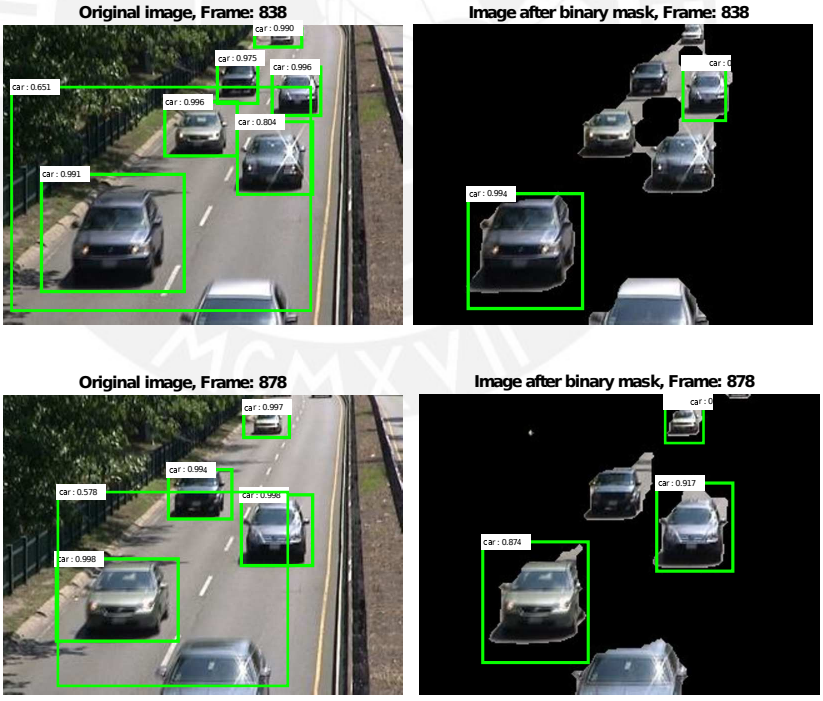


Figure 15: Classification Sample of frames 838 and 878 of the video **highway**. From frame 8338 through frame 878 some of the objects were not classified of even selected as region proposals. This behavior was noted in several sections of the video

	Server			Desktop			Jetson TK1		
Dataset	**Original frame: $\mathbf{d}_k$**	**Masked frame gs-incPCP**	**Masked frame ℓ_1-PCP**	**Original frame: $\mathbf{d}_k$**	**Masked frame gs-incPCP**	**Masked frame ℓ_1-PCP**	**Original frame: $\mathbf{d}_k$**	**Masked frame gs-incPCP**	**Masked frame ℓ_1-PCP**
backdoor	76.1	68.7	**68.7**	145.4	123.2	**123.0**	1024.1	827.4	**812.2**
busStation	81.2	79.3	**76.9**	146.2	135.9	**133.3**	1032.3	958.9	**935.3**
cubicle	82.1	75.1	**71.6**	160.4	151.8	**123.8**	1016.6	890.6	**831.6**
highway	74.5	73.0	**70.4**	178.8	175.1	**127.3**	902.5	867.5	**851.0**
pedestrians	85.0	74.0	**73.0**	224.7	166.7	**121.5**	1085.4	858.2	**847.9**
PETS2006	83.6	**77.9**	80.2	195.1	168.4	**126.8**	983.2	861.3	**842.8**
skating	80.9	**77.8**	81.1	140.5	**134.9**	137.6	919.5	**894.2**	947.2

Table IV: Average Classification times for each video tested of the CDNet Dataset, all times are in milliseconds. It can be noted that the use of any variant of the PCP algorithm for segmentation of the background objects allows a faster classification time, achieving a better performance with the ℓ_1B-PCP variant.

Chapter 5

Discussion

The results from Table II show that independently of the architecture being used, the classification performance remains unchanged as expected. One of the most remarkable results obtained is that in most of the cases the F-measure improved when the sparse image was used to detect and classify the objects with the neural network. The main reason for this is that the neural network finds the features of only the moving objects, instead of all the image, which decreases the False Positives in the classification process. This can be noted in Figure 13 where a water hydrant has been misclassified as a person in the original image while this error was avoided in the Sparse image.

When the main interest is classifying only moving objects in a video, e.g. for video surveillance, traffic control, etc., it is important that the rate of False Positives, i.e. misclassification of objects as the objects of interest, is low. In Figure 13 we can appreciate a water hydrant being misclassified as a person in the original image. This error was appreciated through all the video and thus, decreased the F-measure for the original video classification. In the case of the classification of the Sparse video, since only moving objects are depicted in the images, the misclassification of objects is decreased, giving a better result in means of the F-measure. As can be noted in Table II, this improvement led to an increase of 45% in the F-measure. shows the classification of the frame 595 of the "pedestrians" dataset. Here we can observe that in the case of the original image, a water hydrant was classified as a person, this error was persistent through all the video, decreasing the F-measure. In the case of the "skating" dataset, the gs-incPCP classification show a slightly better performance than the classification of the original images, and the ℓ_1B-PCP showed a worst performance. This is due to the nature of the video, where there is presence of artifacts in the image, i.e. snow falling, and the low contrast of the background, which influenced in both PCP algorithms.

From Table II we can appreciate that two datasets, "cubicle" and "highway", obtained a better F-measure. In the case of "cubicle" the people walking by stand still for certain periods of times, and the PCP algorithm considers them as part of the background as can be seen in Figure 14, this problem is recurrent over all the video and thus decreases the performance. In the case of the "highway" video, it can be noted that no regions were proposed for some objects although these have good contrast and have enough visible features to be classified. In Figure 15, we can note some of the cars not being classified, or even recognized as a region proposal, being this an issue of the Faster R-CNN model. For both videos, some of the objects lack good contrast with the background and lose some necessary features for the classification.

The classification time also was improved when Images obtained from the sparse component of the PCP algorithm were used with the neural network. This improvement can be observed in Table IV. We can note that the classification task performed on the **Server** had an improvement in the classification time that ranges from 2% to 13%. For the images classified in the **Desktop**, the improvement ranged from 2% to 25% and in the embedded system, **Jetson TK1**, the reduction in the classification time ranges from 2% to 21%.

It is worth to mention that the implementation of the Faster R-CNN model used is not optimized to perform sparse operations. Nevertheless, as can be seen in Figure 16, the memory consumption of the sparse images is substantially lower than the corresponding consumption of the original images, obtaining a memory reduction of up to 7x. In the case of the original sequences, the memory consumption remained constant through all the frames, while the memory usage of the sparse images depends completely of the amount of objects in each frame.

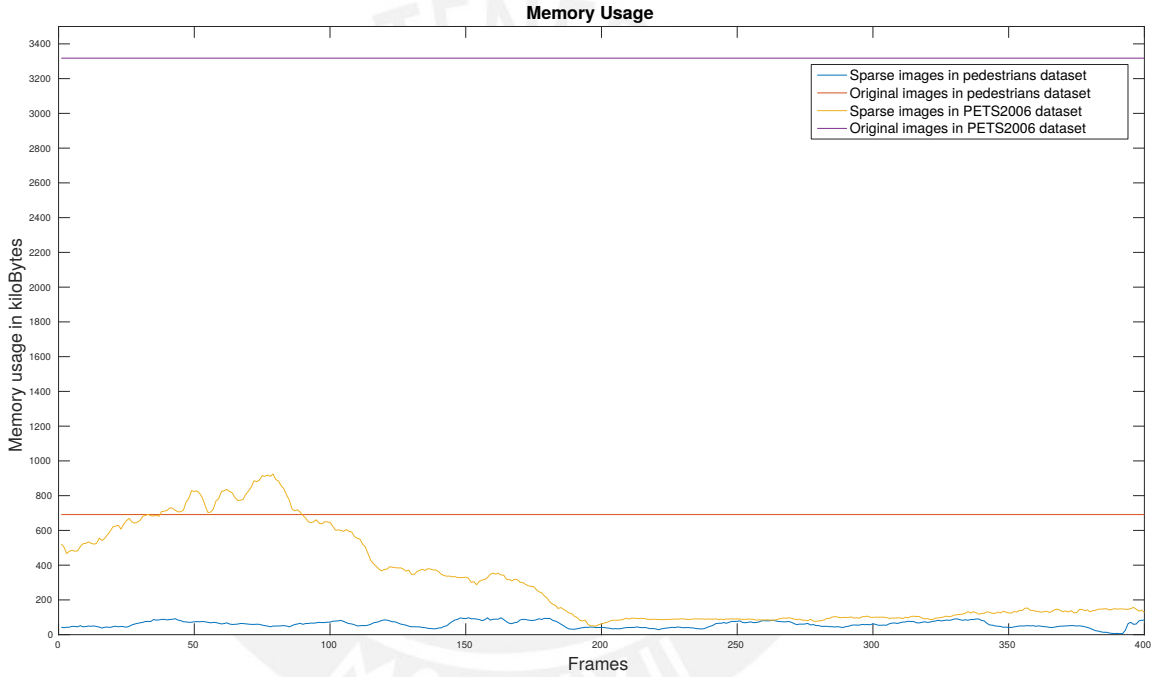
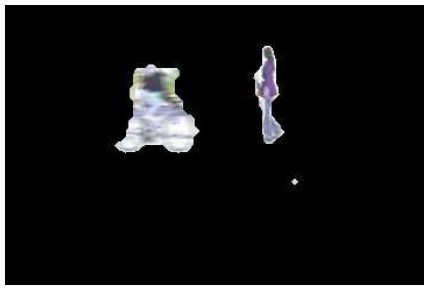


Figure 16: Memory usage for the “pedestrians” and “PETS2006” sequences. As can be noted, the sparse images have a considerable lower memory consumption when read as sparse matrix. See Appendix A for results corresponding to other test videos.

(a) Frame 477 from the original pedestrians video



(b) Frame 477 from the sparse pedestrians video

(c) Frame 690 from the original pedestrians video

(d) Frame 690 from the sparse pedestrians video

(e) Frame 149 from the original PETS2006 video

(f) Frame 149 from the sparse PETS2006 video

(g) Frame 468 from the original PETS2006 video

(h) Frame 468 from the sparse PETS2006 video

Figure 17: Sample frames of the "pedestrians" and "PETS2006" videos. As can be seen, those frames with more objects in the sparse representation have a higher memory consumption than those that have less or none objects. In the case of the original videos, the memory usage remained constant through all the video.

Chapter 6

Conclusions

For certain applications it is important to classify the moving objects in a video, without taking care of the background, e.g. surveillance, traffic control, etc. It was shown that Convolutional Neural Networks can provide an accurate classification of images, achieving state-of-the-art results, however, when the objective is to classify moving objects, current CNN models, such as the Faster-RCNN model, get a low performance due to different reasons, such as misclassification of static objects, grouping of objects into one bounding box, etc. To overcome this problems, we have shown that applying a pre-processing step to segment the moving objects, specifically using the Incremental Principal Component Pursuit algorithm, we can obtain better results.

From the resource consumption point of view, the proposed model could potentially be beneficial for mobile applications, this is due to the sparsity nature of the images after the pre-processing step as well as the reduction of regions or objects to be classified in the image, which leads to an improvement of the classification time, as can be seen in Table IV, as well as memory usage curves shown in Appendix A.

Alternatively, to further improve the behavior of the model, we think that the usage of more specific linear algebra libraries focused on solving sparse algorithms could improve the classification time. Also, a Neural Network that could provide the sparse component alongside with the bounding boxes of the moving objects could increase the performance of the classification task while reducing the classification time.

Appendix A

Memory Usage

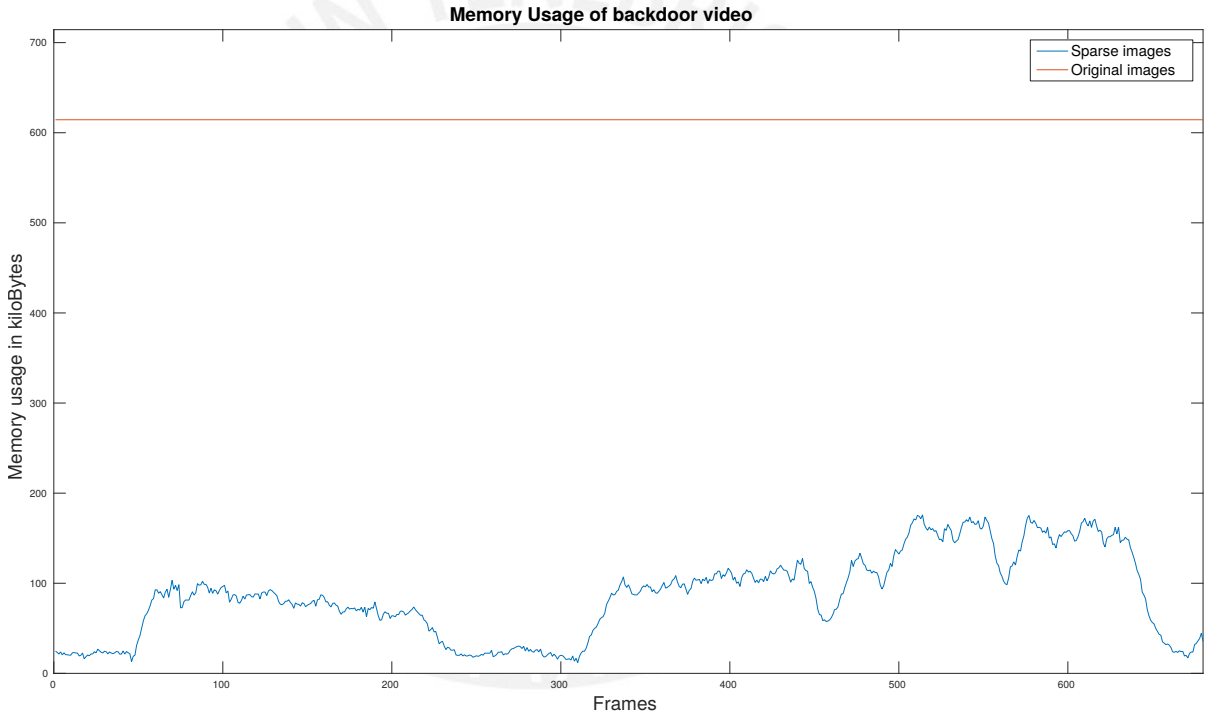


Figure 18: Memory usage for the backdoor sequence

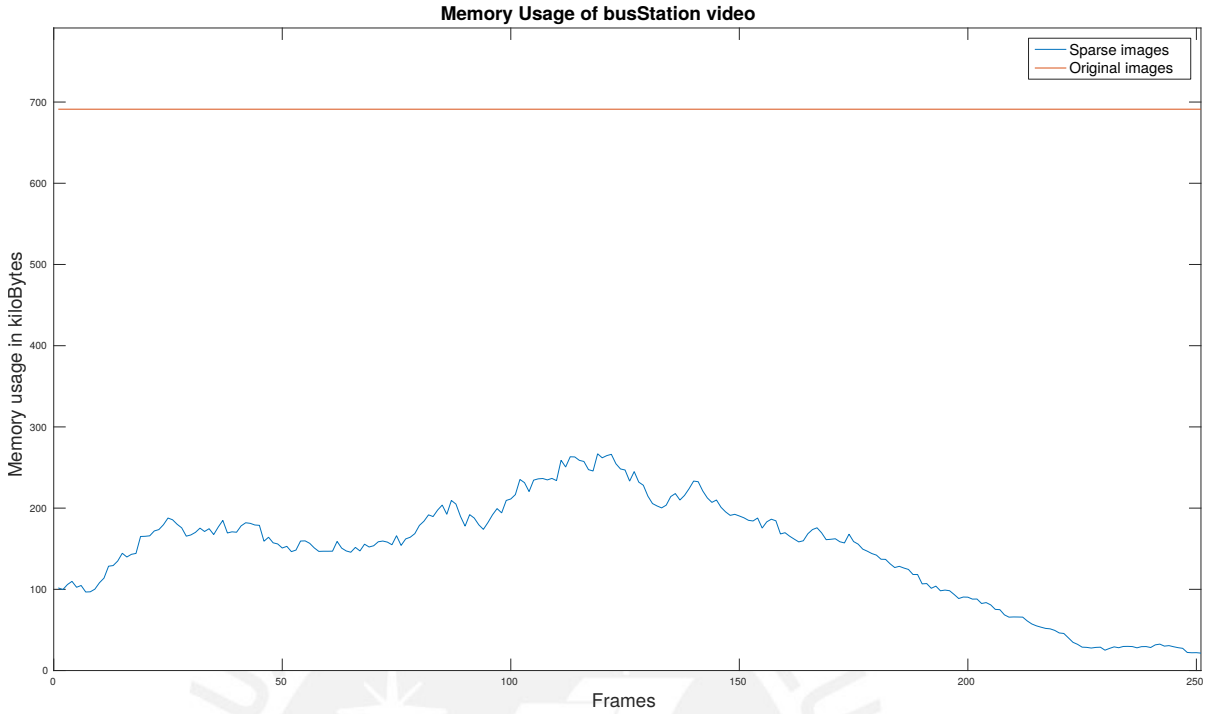


Figure 19: Memory usage for the busStation sequence

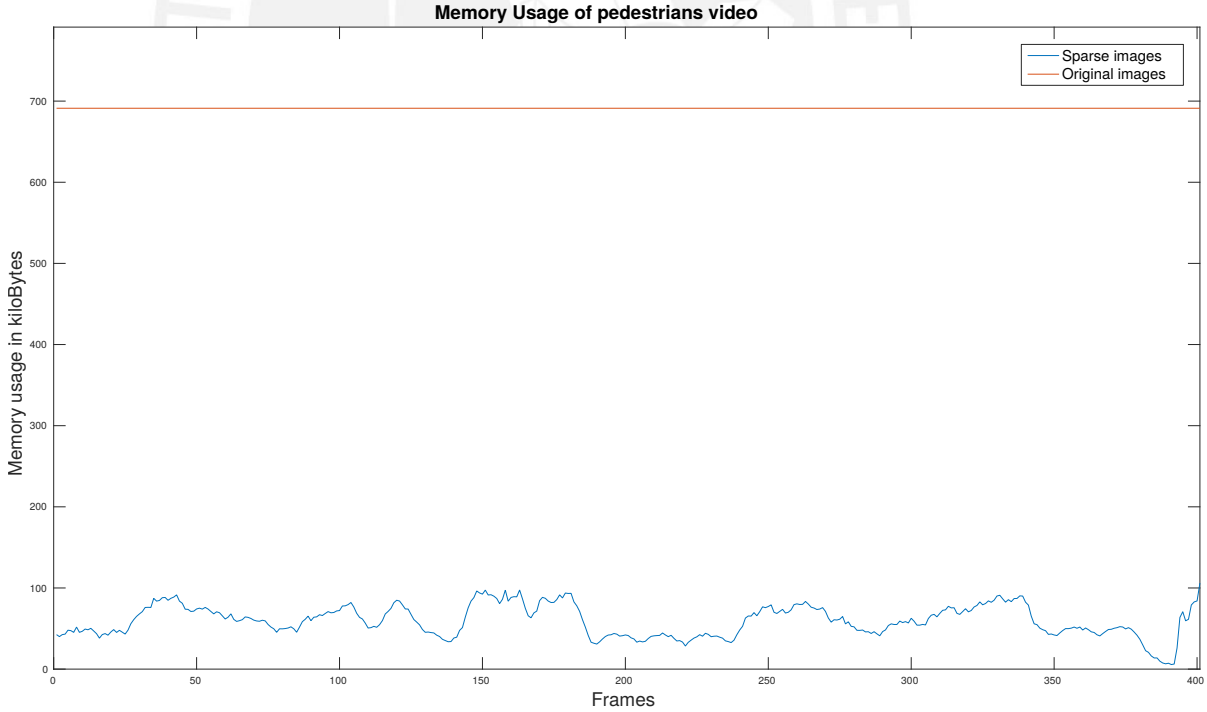


Figure 20: Memory usage for the pedestrians sequence

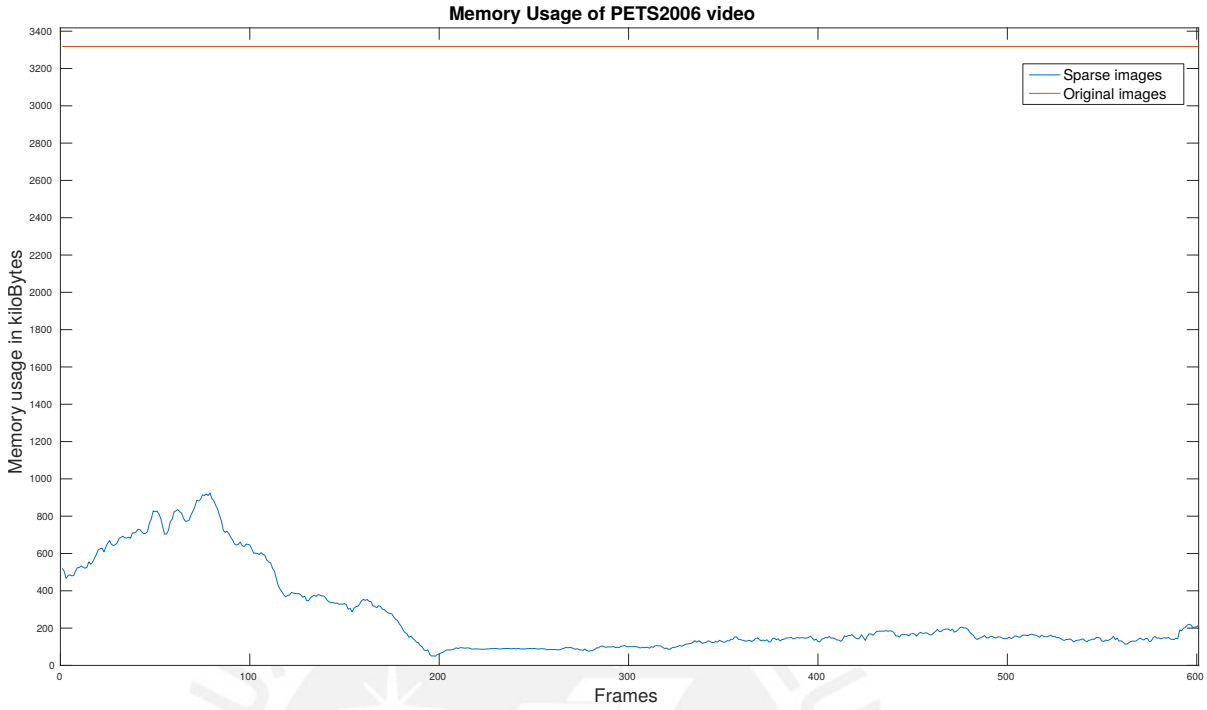


Figure 21: Memory usage for the PETS sequence

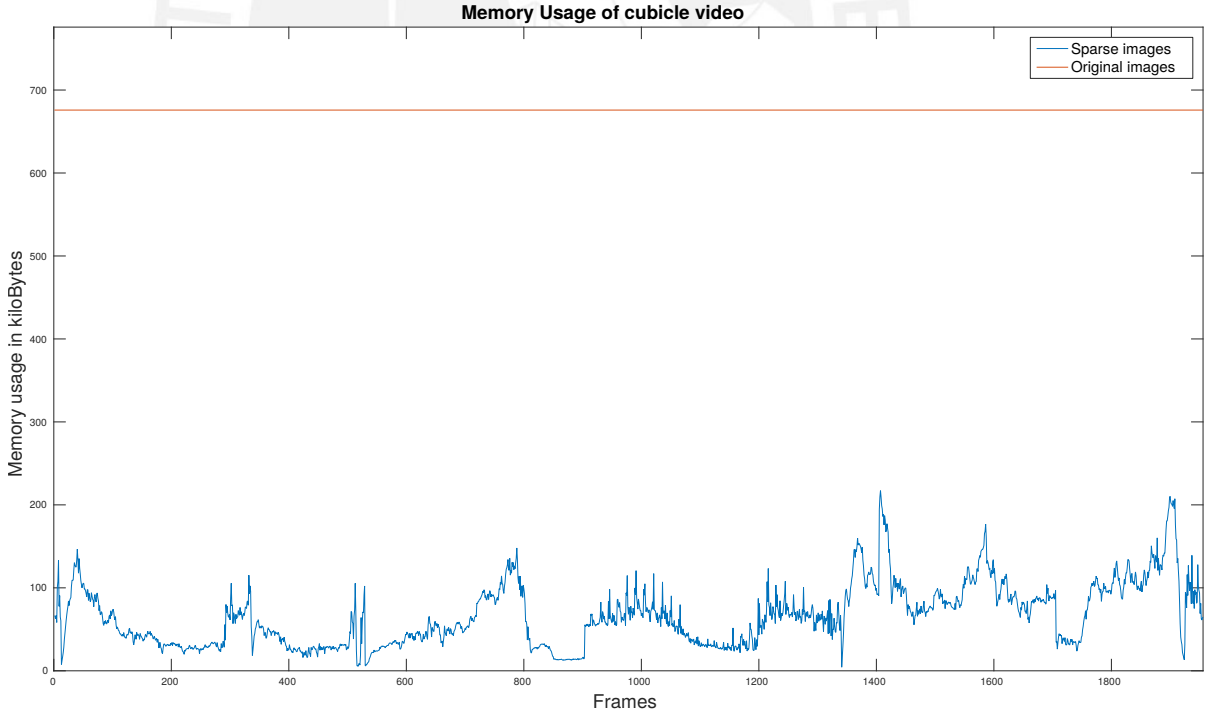


Figure 22: Memory usage for the cubicle sequence

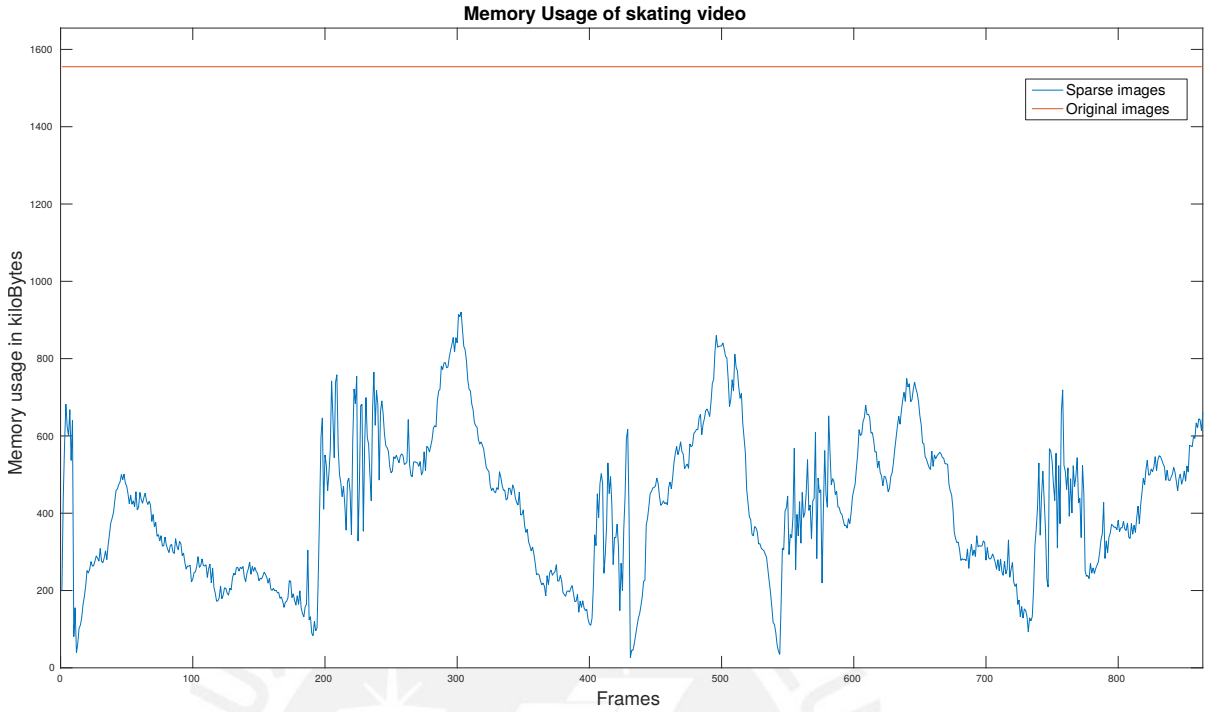


Figure 23: Memory usage for the skating sequence

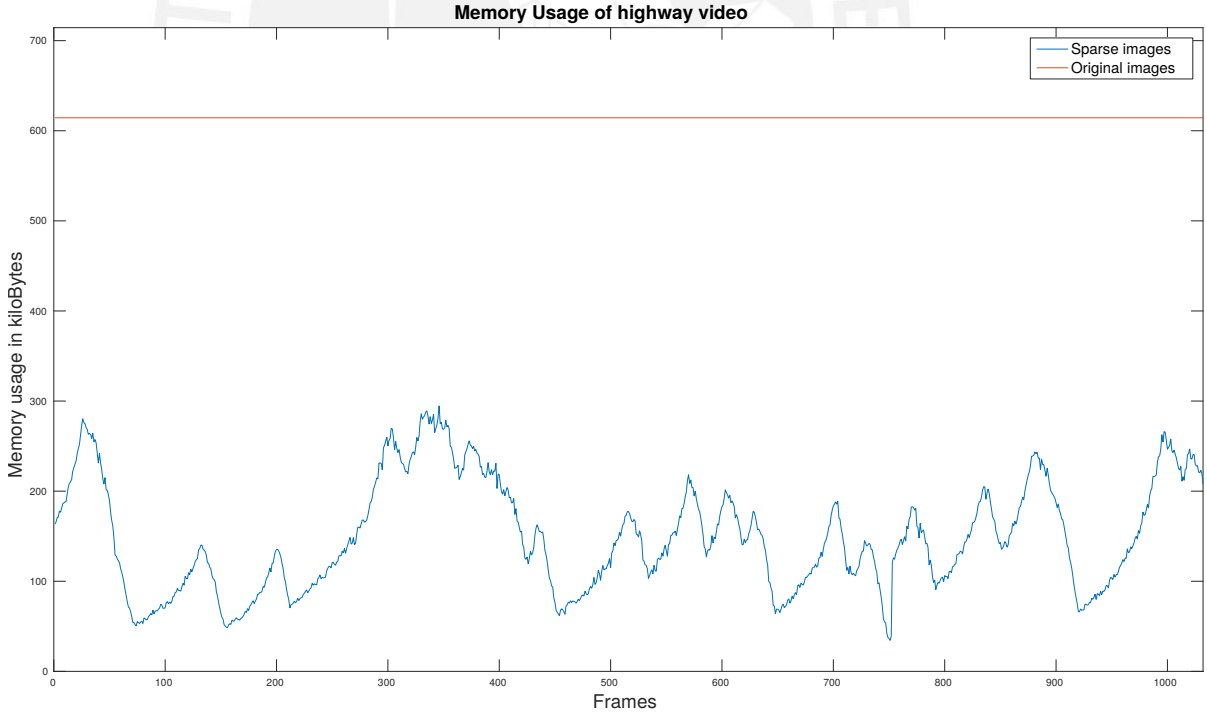


Figure 24: Memory usage for the highway sequence

References

[1] Thierry Bouwmans and El Hadi Zahzah, "Robust PCA via Principal Component Pursuit: A review for a comparative evaluation in video surveillance," *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014.

[2] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, *Microsoft COCO: Common Objects in Context*, pp. 740–755, Springer International Publishing, Cham, 2014.

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded up robust features," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.

[6] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2548–2555, 2011.

[7] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pp. 886–893, 2005.

[8] David G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, pp. 1150–, 1999.

[9] Alex Krizhevsky, Ilya Sutskever, and Hinton Geoffrey E., "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems 25 (NIPS2012)*, pp. 1–9, 2012.

[10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 91–99. Curran Associates, Inc., 2015.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[13] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, and Xiaoou Tang, "DeepID-Net: Deformable deep convolutional neural networks for object detection," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 2403–2412.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, `http://www.deeplearningbook.org`.

[15] Jasper Uijlings, Koen van de Sande, Theo Gevers, and Arnold Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[16] Joao Carreira and Cristian Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1312–1328, July 2012.

[17] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, Nov 2012.

[18] Lawrence Zitnick and Piotr Dollár, *Edge Boxes: Locating Object Proposals from Edges*, pp. 391–405, Springer International Publishing, Cham, 2014.

[19] Yann LeCun, Bernard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge 2016," `image-net.org/challenges/LSVRC/2016/results`, 2016, [Online; accessed 18-April-2017].

[21] Maxwell D. Collins and Pushmeet Kohli, "Memory bounded deep convolutional networks," *CoRR*, vol. abs/1412.1442, 2014.

[22] Ian Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.

[23] Aapo Hyvärinen and Erkki Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, May 2000.

[24] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, Jan 1997.

[25] Charles P. Schofield Nigel J. B. McFarlane, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, no. 3, 1995.

[26] Mark Hallenbeck Jianyang Zheng, Yinhai Wang, Nancy Nihan, "Extracting Roadway Background Image: Mode-Based Approach," *Transportation Research Record Journal of the Transportation Research Board*, vol. 1944, 2006.

[27] Horng-Horng Lin, Tyng-Luh Liu, and Jen-Hui Chuang, "A probabilistic svm approach for background scene initialization," in *Proceedings. International Conference on Image Processing*, June 2002, vol. 3, pp. 893–896 vol.3.

[28] Nuria Oliver, Barbara Rosario Rosario, and Alex Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000.

[29] Dubravko Culibrk, Oge Marques, Daniel Socek, Hari Kalva, and Borko Furht, "Neural Network Approach to Background Modeling for Video Object Segmentation," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, 2007.

[30] Lucia Maddalena and Alfredo Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, 2008.

[31] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Adv. in Neural Inf. Proc. Sys. (NIPS) 22*, 2009, pp. 2080–2088.

[32] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the Association for Computing Machinery (JACM)*, vol. 58, no. 3, pp. 11, 2011.

[33] Huan Xu; Constantine Caramanis; Sujay Sanghavi, "Robust PCA via Outlier Pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 5, 2012.

[34] Paul Rodriguez and Brendt Wohlberg, "A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3414–3416.

[35] Paul Rodriguez and Brendt Wohlberg, "Incremental Principal Component Pursuit for Video Background Modeling," *Journal of Mathematical Imaging and Vision*, vol. 55, no. 1, pp. 1–18, 2016.

[36] Paul Rodriguez and Brendt Wohlberg, "Fast principal component pursuit via alternating minimization," in *IEEE Int'l. Conf. on Image Proc.*, Sept. 2013, pp. 69–73.

[37] Chenlu Qiu and Namrata Vaswani, "Support predicted modified-CS for recursive robust principal components pursuit," in *IEEE Int'l Symposium on Information Theory*, 2011.

[38] Han Guo, Chenlu Qiu, and Namrata Vaswani, "An Online Algorithm for Separating Sparse and Low-Dimensional Signal Sequences From Their Sum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, 2014.

[39] Jun He, Laura Balzano, and Arthur Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1568–1575, 2012.

[40] Florian Seidel, Clemens Hage, and Martin Kleinsteuber, "pROST: a smoothed lp-norm robust online subspace tracking method for background subtraction in video," *Machine Vis. and Apps.*, vol. 25, no. 5, pp. 1227–1240, 2014.

[41] J. Xu, V. Ithapu, L. Mukherjee, J. Rehg, and V. Singh, "GOSUS: Grassmannian online subspace updates with structured-sparsity," in *IEEE Int'l, Conf. on Comp. Vis.*, Dec. 2013, pp. 3376–3383.

[42] Guangcan Liu, Zhouchen Lin, and Yong Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Johannes Fürnkranz and Thorsten Joachims, Eds. 2010, pp. 663–670, Omnipress.

[43] Zhouchen Lin, Minming Chen, and Yi Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," *arXiv:1009.5055*, p. 23, 2013.

[44] Paul Rodríguez and Brendt Wohlberg, "Ghosting suppression for incremental principal component pursuit algorithms," in *Proceedings of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Washington, D.C., USA, Dec. 2016.

[45] Paul Rodriguez and Brendt Wohlberg, "An incremental principal component pursuit algorithm via projections onto the ℓ_1-ball," in *submitted, International Congress on Electronics, Electrical Engineering and Computing,*, 2017.

[46] Paul Rodriguez, "A parallel algorithm for projections onto the ℓ_1 -ball," in *submitted, International Workshop on Machine Learning for Signal Processing,*, 2017.

[47] Yann Le Cun, Ofer Matan, Bernhard Boser, John Denker, Don Henderson, Richard E. Howard, Wayne Hubbard, Larry Jacket., and Henry S. Baird, "Handwritten zip code recognition with multilayer networks," in *[1990] Proceedings. 10th International Conference on Pattern Recognition*, 1990, vol. ii, pp. 35–40.

[48] Yi-Tong Zhou and Rama Chellappa, "Computation of optical flow using a neural network," in *IEEE International Conference on Neural Networks*, 1988, vol. 1998, pp. 71–78.

[49] David H. Hubel and Torsten N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[50] David H. Hubel and Torsten N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[51] David H. Hubel and Torsten N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[52] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[53] Ofer Matan, Christopher JC Burges, Yann LeCun, and John S Denker, "Multi-digit recognition using a space displacement neural network," in *Advances in neural information processing systems*, 1992, pp. 488–495.

[54] David Eigen, Dilip Krishnan, and Rob Fergus, "Restoring an image taken through a window covered with dirt or rain," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 633–640.

[55] Matthew D Zeiler and Rob Fergus, *Visualizing and Understanding Convolutional Networks*, pp. 818–833, Springer International Publishing, Cham, 2014.

[56] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[57] Paul L. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, pp. 2083–2096, 2001.

[58] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 393–400.

[59] Gustavo Silva and Paul Rodriguez, "Jitter invariant incremental principal component pursuit for video background modeling on the TK1," *Conference Record - Asilomar Conference on Signals, Systems and Computers*, vol. 2016-February, no. 1, pp. 1403–1407, 2016.