Conteo de vehículos a partir de vídeos usando machine learning

Juan Sebastián Navarro Ávila

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2020

Conteo de vehículos a partir de vídeos usando machine learning

Juan Sebastián Navarro Ávila

Tesis presentada como requisito parcial para optar al título de:
Magíster en Ingeniería - Ingeniería de Sistemas y Computación

Director:
Ph.D. Cesar Augusto Pedraza Bonilla

Línea de Investigación:
Vision por computador
Sistemas Inteligentes de Transporte

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2020

Dedicatoria

A mi madre y a mi hermana

Por su apoyo incondicional y consejos durante el duro periodo que termina.

Resumen

Este trabajo presenta un framework para el conteo de vehículos a partir de videos, utilizando redes neuronales profundas como detectores. El framework tiene 4 etapas: preprocesamiento, detección y clasificación, seguimiento y post-procesamiento. Para la etapa de detección se comparan varios detectores de objetos profundos y se proponen 3 nuevos basados en Tiny YOLOv3.

Para el rastreo, se compara un nuevo rastreador basado en IOU con los clásicos: Boosting, KCF, TLD, Mediaflow, MOSSE y CSRT. La comparación se hace en base a 8 métricas de seguimiento multiobjeto sobre el conjunto de datos del Bog19.

El conjunto de datos Bog19 es una colección de videos anotados de la ciudad de Bogotá. Las clases de objetos anotados incluyen bicicletas, autobuses, coches, motos y camiones. Finalmente el sistema es evaluado para la tarea de contar vehículos en este conjunto de datos.

Para la tarea de conteo, las combinaciones de los detectores propuestos y los rastreadores Medianflow y MOSSE obtienen los mejores resultados. Los detectores encontrados tienen el mismo desempeño que los del estado del arte pero con una mayor velocidad.

Palabras clave: vehículo, análisis de video, aprendizaje de maquina, visión por computador, aprendizaje profundo, detección de objectos, rastro de objetos..

Abstract

This work presents a framework for vehicle counting from videos, using deep neural networks as detectors. The framework has 4 stages: preprocessing, detection and classification, tracking, and post-processing. For the detection stage, several deep object detector are compared and 3 new ones are proposed based on Tiny YOLOv3.

For the tracking, a new tracker based on IOU is compared against the classic ones: Boosting, KCF, TLD, Mediaflow, MOSSE and CSRT. The comparison is based on 8 multi-object tracking metrics over the Bog19 dataset.

The Bog19 dataset is a collection of annotated videos from the city of Bogota. The annotations include bicycles, buses, cars, motorbikes and trucks. Finally, the system is evaluated for the task of vehicle counting on this dataset.

For the counting task, the combinations of the proposed detectors with the Medianflow and MOSSE trackers obtain the best results. The founded detectors have the same performance as those of the state of the art but with a higher speed.

Key words: vehicle, video analysis, machine learning, computer vision, deep learning, object detection, object tracking.

Content

List of figures

List of Tables

1 Introduction

In an urban environment, the monitoring task covers more traffic behaviors, more road users and objects on the images increase their variety in comparison with a highway environment [1]. Forbidden turns, heavy traffic or illegal parking can be found. The motorbikes, bicycles and pedestrians are the most common road users and non-transport related-objects may appear, increasing the difficulty of analyzing the urban traffic behavior.

An important traffic statistic is the quantity and direction of vehicles traveling in a determined area. Usually, magnetic sensors have been installed on the road for counting, but at a high cost [1]. With videos, the data extraction can be done at a lower cost, which has generated commercial solutions like one in [2] and the development of several research works.

In 2015, the traffic authority of Bogotá city ("Secretaría de Movilidad", SDM) launched the Traffic Management Center, which monitors and manages traffic at 350 points in the city [3]. Its equipment includes radars, surveillance cameras and sensors on the road network for counting. These counting sensors are few and only detect when an object passes over them without classifying it.

The "Programa de Investigación en Tránsito" (PIT) of the Universidad Nacional de Colombia was contracted by the SDM to carry out traffic monitoring. It takes measurements of speed, counts, flows and occupation by direct observation of the road. This manual process rarely used because it is very costly.

The information provided by the cameras already installed in the city would allow constant monitoring of the traffic, even though an automatic and cost-effective method to process the image sequences is needed. This method should overcome the challenges of working with these sources and obtain the necessary traffic data.

1.1. Problem identification

The process of detecting and following up vehicles using video surveillance cameras becomes challenging due to the different angles in which these cameras are located, with respect to the vehicles to be monitored. Therefore, detection from these videos is difficult and requires robust methods tolerant to angle changes.

Consequently, the problem to be solved is to determine the characteristics of a robust vehicle counting method that must be tolerant to the changes of the vehicle perspective with respect to the camera location. For this purpose, only machine learning techniques will be used for the detection of vehicles.

The guiding question for this work is: How to machine learning-based detection and classification models can be adapted to the problem of counting vehicles in video sequences taken from different perspectives?

1.2. Objectives

1.2.1. General objective

To develop a system for vehicle counting that uses video sequences as input with different perspectives and is based on computer vision and machine learning techniques.

1.2.2. Specific objectives

1. To determine the characteristics that define the robustness of a detection method when the perspective of the vehicle changes in each video sequence.

2. To design and implement a detection method for vehicles on images from surveillance cameras.

3. To design and implement a method for tracking vehicles in video footage taken by surveillance cameras.

4. To develop a software system that integrates detection and tracking methods for vehicle

counting in video sequences taken by surveillance cameras.

5. To evaluate the system on a set of videos collected from monitoring done by the *Secretaria de Movilidad* of Bogotá city.

2 Theoretical framework

2.1. Visual surveillance

The video sensors are an important source of data for the Intelligent Transportation Systems (ITS) [3]. There are 4 main reasons for their use: 1) the people are used to visual information; 2) the video sequences can cover a wide information spectrum; 3) they are easily installed, operated and maintained, and 4) the price-performance ration has improved with the time. The obtained image quality depends on the environment, location and camera characteristics.

The traffic parameters extraction from video has 3 phases: vehicle detection and recognition, tracking and analysis [4]. The detection phase establish if there are interesting objects on the visible area and separates them from the image background. In the tracking phase, the vehicle location is estimated for each frame of the video and the trajectory is build. Finally, the result is analyzed to extract some parameters like: velocity, number of vehicles, traffic density and accident information.

2.2. Image processing

The digital image transformations are fundamental for prepare the input for the analysis process. Some of the most popular image processing methods are: binarization, RGB to gray scale, noise filtering and down sampling. (Table **2-1**).

Table 2-1: Image preprocessing methods.

Method	Description	Advantages	Disadvantages
Binarization (Black and white)	Transforms a color image to a black and white only image, with and specific threshold.	If the threshold is appropriate the next image process is greatly simplified	There is a huge loss of information.
RGB to Gray	Each frame is converted to a gray scale, the color channels are merged to one.	The variation from the color diversity is removed.	If the posterior phases are based on the image intensity, can have difficulties when the vehicle has the same intensity as the road [4].
Noise filtering	Reduction of the aleatory values in image generated from the capture setup.		
Down-sampled	Reduction of the image resolution [5].	Increase the speed of processing.	Small objects may be lost.

2.3. Object detection

The object detection problem is more general than the image classification one [7]. In the image classification, the objective is determine the classes or categories that the entire image or the main object in the image belongs. In object detection, a class and the location on the image for each object is determined.

The literature about detection about object detection can grouped the according with the approximation, bottom-up or top-down [1]. The bottom-up approximation detects and classifies the object's components first, then the object area in the image is identified based on the presence of components. On the other hand, the top-down groups the pixels that represent an object and this model is propagated by the system. Inside the top-down approximation can be localized the methods based on movement and features.

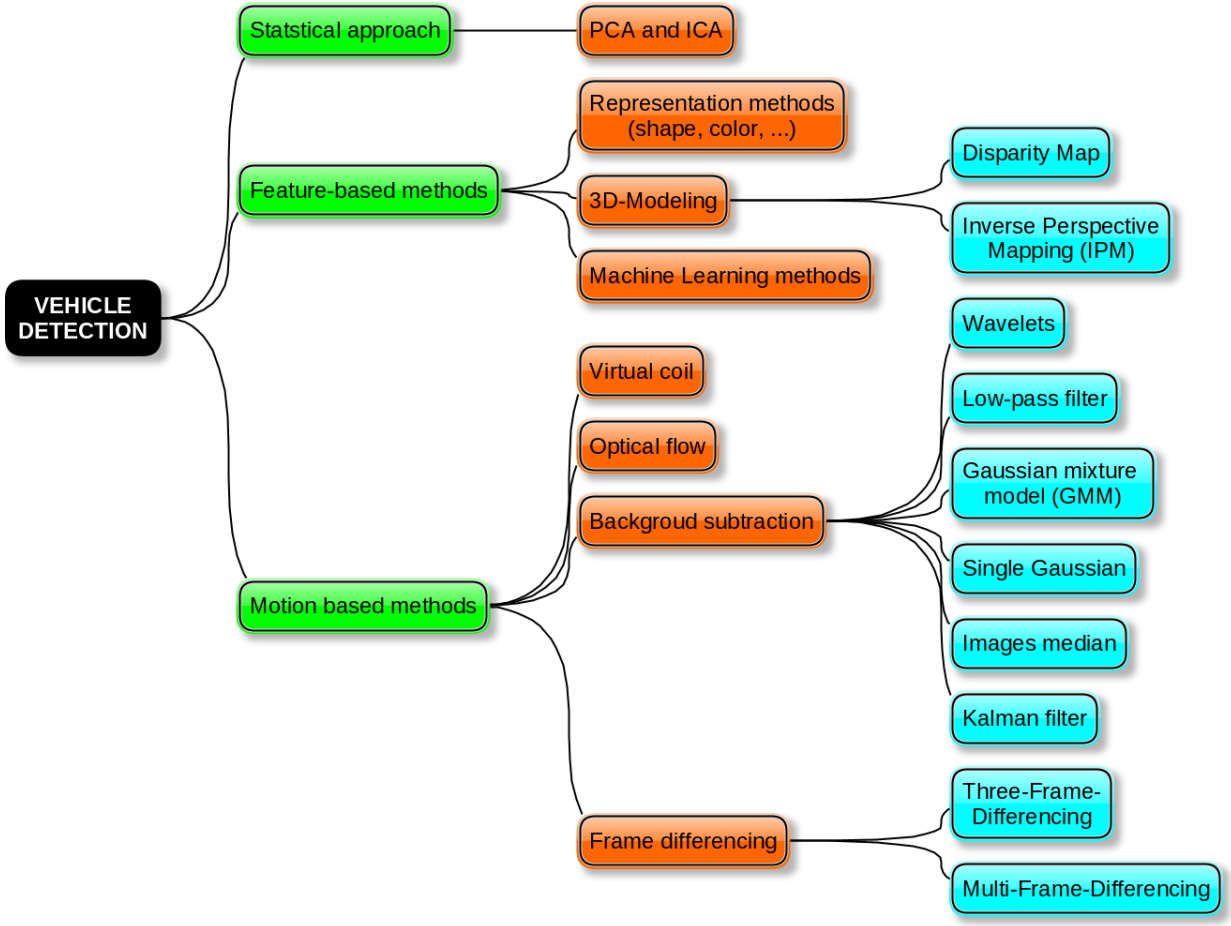

Figure 2-1: Detection methods.

The methods based on movement assume that the main feature of the vehicles is been moving [4]. However, the moving object can be other than a vehicle, and the cars are not always in movement. The methods based on features use the visual information of the object, like color and shape, to create models. These methods are capable of detect stationary vehicles and even recognize them.

Table 2-2: Detection methods description, advantages and disadvantages.

Method	Description	Advantages	Disadvantages
Featured-based methods	Methods based on visible features of the vehicles like color, texture, shape and others calculate from the image [6].	Can detect and recognize moving and parked vehicles [6]. They are suitable when complex distortions are present on the background.	It's difficult discriminate the features of near vehicles.
Representative approaches	The method use information from the vehicle characteristics or its parts, like symmetry, color, edges, contour, texture, shadow [6]. Some relevant parts are the lights, windshield or tires.		The computational complexity depends on the combination of chosen characteristics [6].
3d Modeling	The method build a 3D model from the vehicle to use as search reference [6]. Disparity map and Inverse Perspective Mapping (IPM) are the main used methods.	It allows recover precise tracks [7]. The candidate vehicle must only be compare with a finite set of prototypes.	It's difficult obtain an exact 3D model, specially for moving vehicles [6]. A single model can't be used for all vehicle types. The models may be too simple to use in high resolution images.
Machine learning	Use machine learning techniques to generate a discriminative classifier, from training data, to label unseen images [6].		A big dataset is required for training and testing.

Continuation of Table **2-2**

Method	Description	Advantages	Disadvantages
Motion-based methods	Exploit the moving nature of the vehicles for the detection task [6]. These methods had been frequently used in video surveillance.	Adequate when the background is relative fixed and most of the moving objects in the scene are vehicles [6].	No always a vehicle is moving and a moving object is not always a vehicle [6]. The stationary vehicles can be absorbed by the background as the time pass [8].
(Adaptive) Background subtraction	It calculates the difference pixel by pixel between the current image and the background reference image, given a dynamic or static threshold, to determine the foreground. Exists several methods to build the background image without prior knowledge like: image median, Single Gaussian, Gaussian mixture model (GMM), low-pass filter, Kalman filter and Wavelets [6].	Low computational cost, very suitable for real time processing [6]. It can tolerate changes in light and weather conditions [7].	The updating process for the background reference image can add noise. It requires a background reference without moving vehicles and has problems with occlusions [7].
Optical flow	Estimates the pixel movement based on the temporal changes and their correlation in the image sequence [6].	Obtains information from the vehicle movement [6].	It uses an iterative algorithm consuming many time and has a poor performance when noise is present [6]. For real time processing needs a special hardware setup.

Continuation of Table **2-2**			
Method	**Description**	**Advantages**	**Disadvantages**
Frame differencing	It calculates the difference between consecutive frames at pixel level, with a given threshold. It's preferable use more information than just the last frame, for example: Three-Frame Differencing or Multi-Frame Differencing [6].	It's very fast and adequate for dynamic changes on the background [6].	It's difficult detect multiple, very fast or very slow objects [6]. It doesn't handle well the noise, abrupt changes of illumination or periodic movements in the background like trees [4, 6].
Virtual coil	It set a line or region of interest to watch. When a object pass through the region and the image changes more than a given threshold is considered as a vehicle [6].	The algorithm has low computational cost and its flexibility made it viable for commercial detection systems [6].	Like with the physical coil, the information is limited to the region of interest, discarding the rest of the image [6].

2.3.1. Machine learning

The feature based object detectors can be classified in 3 groups: Classic, One-stage and Two-stage [9]. Some of the classic ones are Convolutional Neural Networks [10], Viola and Jones [11] and HOG [12]. The sliding-window approach was the leading detection paradigm in classic computer vision, in which a classifier is applied on each cell of a dense image grid. If the cell is classified as containing and object, the cell becomes the bounding box for the detected object.

Deep learning [5] is one of the latest advances in the field of object detection, thanks to the progress of parallel computing hardware and software [13]. It's key component is the multilayered hierarchical data representation in the form of neural networks with more than a few layers. The availability of large data sets, powerful hardware and training methods of

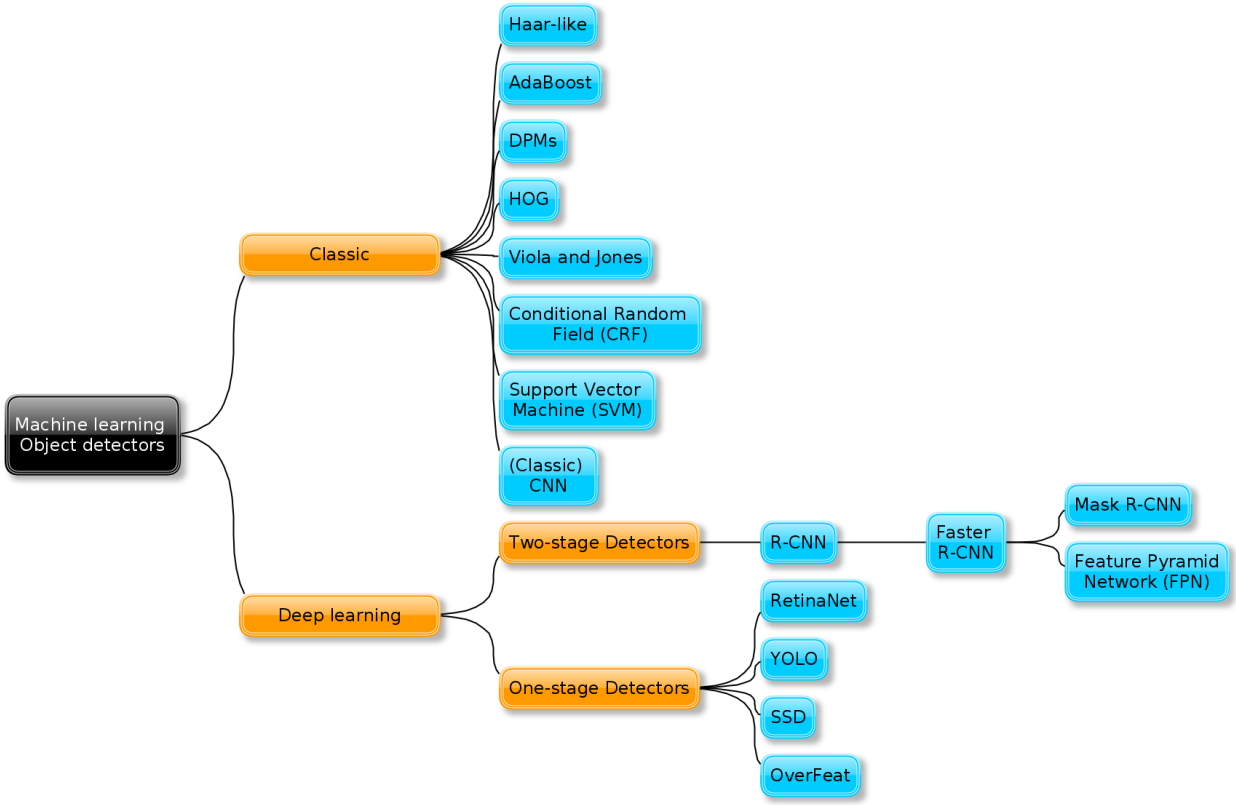

Figure 2-2: Feature based detectors used in Machine Learning.

deep networks awoke a new interest for this area.

After the resurgence of deep learning, the two-stage object detectors came to dominate in object detection [9]. The two-stage detectors generate a set of candidate proposals containing all objects at the first stage, and classify the proposals into foreground or background classes in the second one. R-CNN [14] combines the region proposals with a convolutional neural network as feature extractor and SVMs for region classification, achieving a mAP of 53.3 % on the VOC 2012 dataset. The Faster R-CNN framework [15] integrates the two stages of R-CNN into a single convolution network using Region Proposal Networks (RPN).

The one-stage methods like SSD [16, 17] and YOLO [18, 19, 20] have been tuned for speed but their accuracy is lower than the two-state ones [9]. YOLO uses a single neural network to predict bounding boxes and class probabilities on real time, but it struggles with small or nearby objects and has higher localization errors. It's last version has an mAP of 57,9 % on the COCO dataset, with an inference time of 50ms [20].

RetinaNet is a novel one-stage object detector, that uses a new loss function called focal loss to overcome the previous one-stage and two-stage single-model detectors [9]. The main

problem with the one-stage object detectors is that their accuracy is low compare with the two-stage ones because the class imbalance between foreground and background. Using new loss functions, RetinaNet improves the AP on the COCO data set for a single model.

Table 2-3: Machine learning object detectors.

Method	Description	Advantages	Disadvantages
AdaBoost	Combines weak classifiers into one strong classifier. A weight is assign for each sample, depending on the correctness of the classification [6].	Only a few weak classifiers need to be trained.	
Convolutional Neural Networks (CNN)	Uses artificial neural networks with convolutional layers to detect objects in large datasets [21].	It detects and classify the vehicle, according with the training labels [21]. It doesn't need a pre-training stage compared with Autoencoder (AE) and Restricted Boltzmann Machine (RBM) [13]	The computational cost is proportional to the input image size [21].
One stage detectors	The generation of object proposals locations and classification is done by a single network [9].	High speed on exchange of accuracy [9].	It must process a big quantity of object proposal locations and tends to be more background samples than true objects [9].
Two-stage detectors	The generation of object proposals locations is done by on network and the classification is done by another network [9].	It frequently achieves best accuracy on benchmarks [9].	

Continuation of Table **2-3**			
Method	**Description**	**Advantages**	**Disadvantages**
Histogram of Oriented Gradients (HOG)	It describes the appearance and local shape of and object with a intensity or edge gradient distribution [22].	It operates on localized cells, then invariant to geometric and photometric transformations, except object orientation change [22].	The original HOG is not capable of handle images of different size and aspect, al alternative is SHOG [22].

2.4. Object tracking

The tracking phase is necessary for the object counting, because with the trajectory information the precision of the process can be improved. For example in the work of [23] the tracking information improved the counting results with classification on low resolution images.

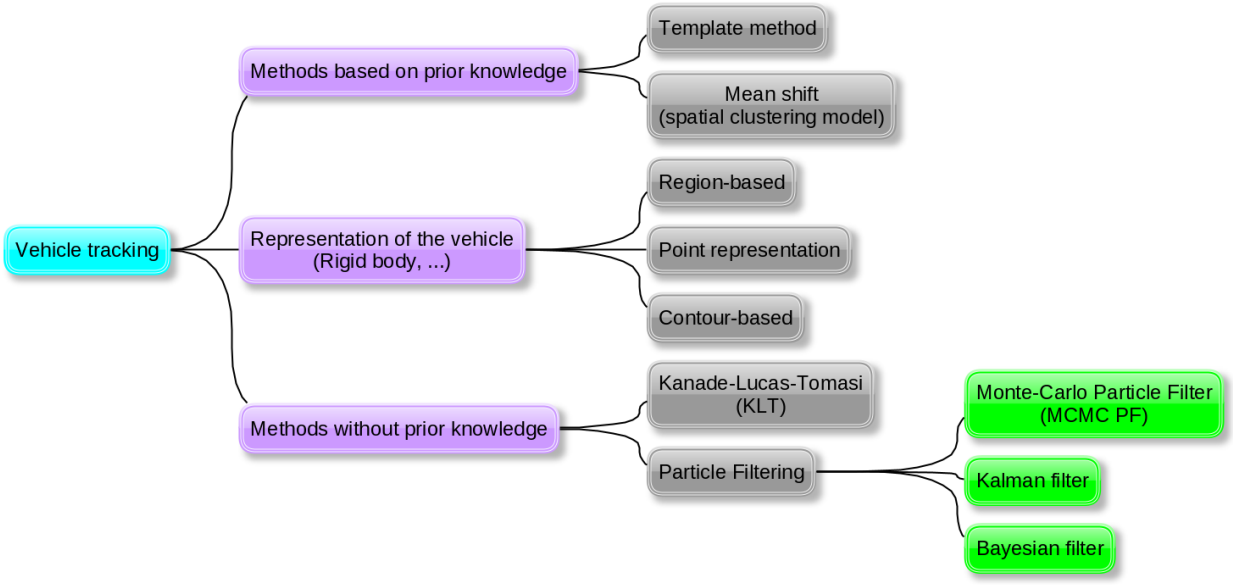

Figure 2-3: Tracking methods.

For this phase several methods had been developed and applied [6]. Some of the more common are the Kalman filter [24] and particle filtering, for which no prior knowledge is required.

Particle filtering overcomes the restriction of a single Gaussian distribution for the Kalman filter.

The tracking of multiple components of an object, bottom-up approximation, had been done in the work [25]. The tracking of vehicles is done with two levels of particle filtering. The first level extracts particles from the appearance features of the components. The second level models a posteriori probability of the constellation. The experiments were conducted against the dataset VIVID-PETS. It reports that a dynamic adaptation of the spacial model rigidity allows the occlusion handling, keeping the relationship between components.

Table 2-4: Machine learning object detectors.

Method	Description	Advantages	Disadvantages
Representation of the vehicle	The vehicle is represented as a rigid body, with dots, area, contour or model [6].		
Point representation		It's adequate for tracking of object with a small area on the image [6].	
Contour-based	It uses a close contour toe represent moving objects. The contour can be continuously and automatically updated [6].		Background edges can interfere with the process.
Methods based on prior knowledge	It uses previous information of the image or the objects to tracking them.		Requires prior knowledge of the shape or other features of the vehicle.
Template Match	It searches a template of the vehicle on the image [6].	It's simple to understand, with high precision and can recognize the vehicle on static or dynamic images [6].	It can take some time for implementation and the traditional algorithm with a rectangular template with gray correlation is computational costly [6].

Continuation of Table **2-4**

Method	Description	Advantages	Disadvantages
Mean Shift	Non-parametric gradient density estimation algorithm, based on a general kernel function [6].	The points with zero probability of gradient density can serve as pattern points in a spacial clustering model [6].	
Particle Filtering	Frequently used to track multiple objects	They can handle non-linearities introduced by occlusions and background clutter [26].	
Kalman filter	Uses a series of measures over time, with noise, and produces more accurate estimations of unknown variables than the ones based on a single measure [6].	It can use simple state variables like position and size, and has been heavily used in automatic traffic surveillance systems [7]. It uses all the historical information and reduces the search range over the image, with a high speed processing [6]. For moving vehicles, it remains stable on illumination changes or occlusion [6].	The performance for relative big vehicles in the image is not satisfactory [6].
Bayesian filter	Builds a posterior density probability from the state variables, using a group of random samples [6].	This approach overcome the restriction of a gaussian distribution of the Kalman filter [6].	

Continuation of Table **2-4**			
Method	**Description**	**Advantages**	**Disadvantages**
Monte-Carlo Particle Filter (MCMC PF)	Uses a Markov Chain Monte-Carlo exploration space to associate data and track objects during a period of time [26].	The quantity of particles is a lineal function of the number of objects to track [26].	The reported capacity of the algorithm is limited to 12 vehicles in [26].

2.5. Public datasets

Exists several public image and video dataset that can be used training and evaluation. For the object detection task: ILSVRC [27], PASCAL VOC [28] and COCO [29].

In Multi-Object tracking the system output is the track for each object. The track has an id, bounding boxes for each frame and the type of object. The following datasets are public for multi vehicle tracking: UA-DETRAC [30] with 100 videos, VisDrone [31] with 288 videos, and KITTI MOTS [32] with 21 training videos.

3 State of art

For the problem of vehicle counting, it has been installed magnetic sensors on the road and cameras on poles [1]. The magnetic sensors are intrusive and come at a high maintenance cost. With videos the information extraction can be done at a lower cost, witch has inspired the realization of several research works and commercial solutions.

There has been an increasing interest for automatic computer vision based analysis of urban traffic activity from videos [1]. The automatic extraction of relevant information can aid human operators observing traffic behavior from video data. The availability of monocular road-side cameras in urban environments, the increasing computer power and development of computer vision algorithms has enabled new applications for Intelligent Transport Systems (ITS).

The traffic parameters extraction from video has 3 phases: vehicle detection and recognition, tracking and analysis [6]. In the detection phase, its established if there are objects of interest on the visible area and are separated from the image background. In the tracking phase, the vehicle location is estimated for each of the video frames and its trajectory is build. Finally, the result is analyzed to extract parameters as: velocity, number of vehicles, traffic density and accident information.

3.1. Vehicle detection

Detection methods can be divided in 2 groups: based on features and movement detection [6]. The motion based methods are usually used on surveillance of scenes where the main nature of the vehicle is moving and are adequate when the background is stationary.

The methods based on features relies on usually on color, texture, shape or other features extracted of the vehicle. These methods can detect stationary cars, recognize the vehicle and are adequate when complex perturbations exists on the background [6]. The methods like HOG [12], Viola Jones [11] and Deep Neural Network [5] are in this group.

The idea of HOG is that object appearance and shape in an image can be described by the distribution of intensity gradients. But HOG is not capable of handle variable size and aspect ratio cars on images, so a method called SHOG was proposed [22].

After the resurgence of deep learning, the CNN came to dominate in computer vision. The works of [33] and [13] present reviews of these algorithms in vision tasks and have shown the best performing experimental results among moderns models. The SSD [16, 17], YOLO [18, 19, 20] and RetinaNet [9] are the top rated single model for object detection.

3.2. Vehicle tracking

Exists two main approaches for establishing the trajectory of an object over time, tracking [34]. The first uses the initial detection and the information of previous frames to estimate the correspondence between object instances, and the second one tracks by object detection on each frame.

In tracking by detection first a object detector is applied on each video frame and the tracker associates these detections to tracks. The quality of the results are limited by the detector performance and the tracker capacity to handle missing detections, false positives and ID switches. The work of [35] reviews the use of deep learning for the Multiple Object Tracking (MOT) task.

The IOU tracker achieves high speed on the UA-DETRAC dataset, assuming that the detections of an object on consecutive frames have high IOU and without using visual information [36]. Extending the IOU tracker with a visual tracker used when no detection satisfies the IOU threshold, reduces the ID switches and fragmentations on the UA-DETRAC and VisDrone datasets [37].

3.3. Visual vehicle counting

The work of [38] was one of the first using image processing to estimate traffic parameters. From captured images with television cameras, mounted in posts, the background is subtracted to detect the vehicles and a signature from the image segment is created when the object passes over the detection line. The trajectory is determined searching the signature on the next frames. It reports that the field test were successful for a lane of traffic.

The work of [39] makes the vehicle counting through the background subtraction and the tracking with the Kalman filter. The background separation is made based on characteristics, taking in account the shadows and the object occlusions. For this, presents a new characteristic called “linearity”, to classify the vehicles according to their size and velocity.

4 Robustness characteristics

In situations involving image recognition, it's required an insensible system to background, position, orientation, illumination and near objects variations, respect to the object of interest [40].

4.1. Scenes

A more robust method is needed to deal with complex traffic scenes such as these.

Table 4-1: Scene types.

Scene	Description	Papers
On-road	The camera is located on or inside the vehicle. Only the nearest objects are visible, at the road level.	[22]
Highway surveillance	A multi-lane roadway is visible, with vehicles only. Traveling in one or several directions. It's a fixed top view of the vehicles, usually far from the camera.	
Lane surveillance	All the visible vehicles travel in the same direction, and the visible roadway is straight.	[4, 41]
Urban	Several vehicles and other objects are visible.	[21]
Night	The recording is at night.	[4]

4.2. Camera

Among the challenges that must be overcome to achieve automatic visual vehicle counting the camera location, type, movement and calibration. Although cameras are usually monocular and static, until an operator takes control, the perspective of the vehicle may change as it moves.

Table 4-2: Camera.

	Property	Description	Papers
Len type	Monocular	The camera has a single lens. This is the most common type.	[4, 41, 22]
	Stereo	The camera has two lenses.	[21]
Camera location	On infrastructure	The camera is anchored in the infrastructure and fixed looking at the traffic.	[4, 41]
	On the vehicle	The camera is mounted inside or on top of the vehicle, captures the vehicles in front.	[21, 22]
Camera movement	Environmental	The camera moves because of the present natural elements, like the wind, without leaving its position. It may loose focus.	[21, 4, 22]
	Automatic	The camera has an automatic movement to cover a wide area.	
	Human directed	A human operator moves the camera to see an area of interest.	
	None		[41]

4.3. Image quality

The chosen properties and changes in the image, due to weather or illumination, can have a severe impact on the object detection results.

Table 4-3: Image quality.

Property	**Value**	**Papers**
Color scale	Color	[22]
	Gray	[4, 41]
	Color and gray	[21]
Image resolution	1280x1024	[22]
	1242x375	[21]
	320x240	[41]
	Unspecified	[4]

4.4. Object Occlusion

The occlusion is one of the main problems to solve of visual object tracking. It's common in the urban scenes and introduces ambiguity in the vehicle detection, this causes erroneous estimations of traffic parameters [42]. The occlusion can be mitigated with cameras installed at higher poles [1], the extra height provides a better viewing angle.

Regardless of the method, occlusion is one of the problems to be considered in the detection phase. For example, in the work of [41] a system was built for the control of traffic-light intersections using a Haar classifier and the Adaboost algorithm for vehicle detection. Although the system was tested using simulations, it was shown that the proposed detection algorithm is sensitive to occlusions greater than 10 % of the visible area of the object.

5 Vehicle counting system

The system has 4 phases: image preprocessing, vehicle detection, tracking and post-processing. In the first phase, each of the video frame is read and scaled. In the object detection phase, a object detector is passed over the frame. The detection results are used as the input for a tracking algorithm. The bounding box for each track are plotted over the frame and saved on a new video. The figure **5-1** shows the described system workflow.

The system follows the top-down approximation for traffic analysis systems described in [1].

The vehicle counting results are calculated based on the tracks. For each active track on the frame an unit is added to the total count by vehicle type.

5.1. Image preprocessing

The input frame is scaled to reduce the number of pixels to process. The target resolution has a minimum side of 600 pixels keeping almost the same aspect ratio, for example a 1920x1080 image will be reduced to 1066x600. The objective with this transformation is control the speed of the system, as the image size has a great impact on object detection performance.

5.2. Vehicle detection

For vehicle detection 4 pretrained object detectors and 3 custom models are used to infer the bounding boxes and class probabilities. The pretrainned detectors are Faster R-CNN architecture with the resnet50 model [15], YOLOv3 [20], Tiny YOLO and RetinaNet [9].

YOLO uses a single CNN to predict class probabilities and bounding boxes [18, 19]. YOLOv3 version has reported an mAP of 57,9 % on the COCO dataset, with an inference time of 50ms [20]. Tiny YOLO is an architecture designed for embedded devices.

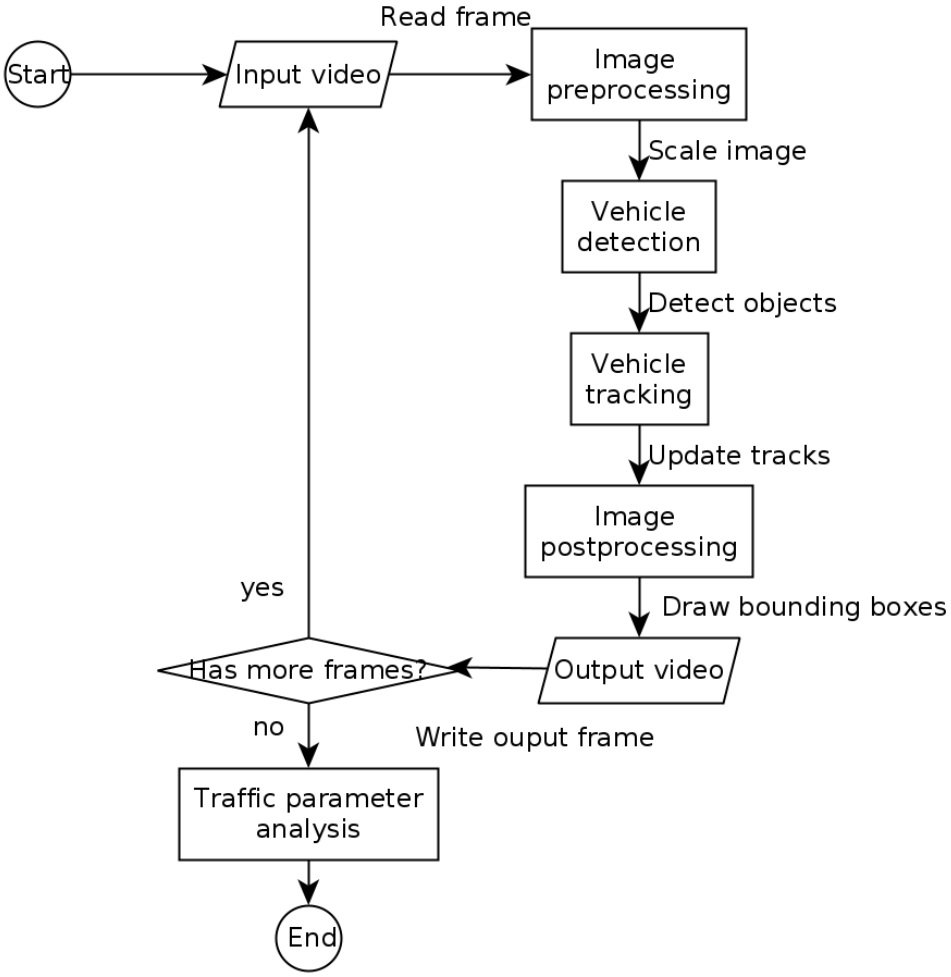

Figure 5-1: System workflow.

The 3 new custom networks VVC1, VVC2 and VVC3, are based on Tiny YOLOv3. The networks are has `DarknetConv2D_BN_Leaky` layers, witch represent a Convolutional 2D layer followed by a Batch Normalization layer and a Leaky ReLU layer with alpha 0,1. At the end of the networks a lambda layer, with no trainable weights, is used for the YOLO loss calculation.

The VVC1 architecture, in the figure **5-2**, was obtained adding a `DarknetConv2D_BN_Leaky` layer at the beginning of Tiny YOLOv3. The expected result is an increase in the abstraction of features from the image.

The VVC2 has a Dropout layer, with a rate of 0.2, after the first max pooling layer, as shown in the figure **5-3**. An increased generalization capability is the expected result of this modification.

The VVC3 replaces the first and second `DarknetConv2D_BN_Leaky` layers with regular convolutional layers, and removes the second max pooling layer and third `DarknetConv2D_BN_Leaky` layer, shown in the figure **5-4**.

A subset of the COCO dataset, called COCOv, is used for training and validation. The new dataset has only images with bicycles, cars, motorbikes, buses or trucks, from the original training and validation sets.

The VVC networks were trained during 10 epochs using the COCOv dataset. The YOLO loss of the networks for each epoch of training and validation is shown in the figure **5-5** and **5-6** respectively.

5.3. Vehicle tracking

A custom tracker based on the IOU tracker [36] is used to build the tracks. The Patient IOU Tracker keeps the tracks that don't satisfied the detection threshold and the minimum track length as inactive tracks. The inactive tracks are not considered as results of the tracking algorithm, but can become active if a detection is assigned to the track based on the IOU threshold. If a track remains inactive for more than p frames the track is discarded, p represents the patience of tracker.

The naive tracker use IoU for the association of each detection with the previous one. The algorithm search for each bounding box the closest track based on the IoU metric. If no track is found, then a new track is started for the type of vehicle. The tracks with no new detections in the last 5 frames are finished.

The parameters for the Patient IOU tracker were determined using a grid-based search as described is table **5-1**. From the 560 parameter combinations, the best is $\sigma_{IOU} = 0{,}5$, $\sigma_h = 0{,}8$, $t_{min} = 4$ and $p = 2$, with a MOTA=-193 % and MOTP=0.28 on the BOG18 dataset using the RetinaNet detector.

Table 5-1: Ranges for Patient IOU tracker parameter grid search.

Variable	Min value	Max value	Step
σ_{IOU}	0.3	0.7	0.1
σ_h	0.5	0.8	0.1
t_{min}	1	4	1
p	2	8	1

5.4. Image post-processing

The system draws the bounding boxes and track id over the scaled frame and it writes an output video. The bounding boxes come from the tracking phase and the track id is a composition with the vehicle type and the number of track.

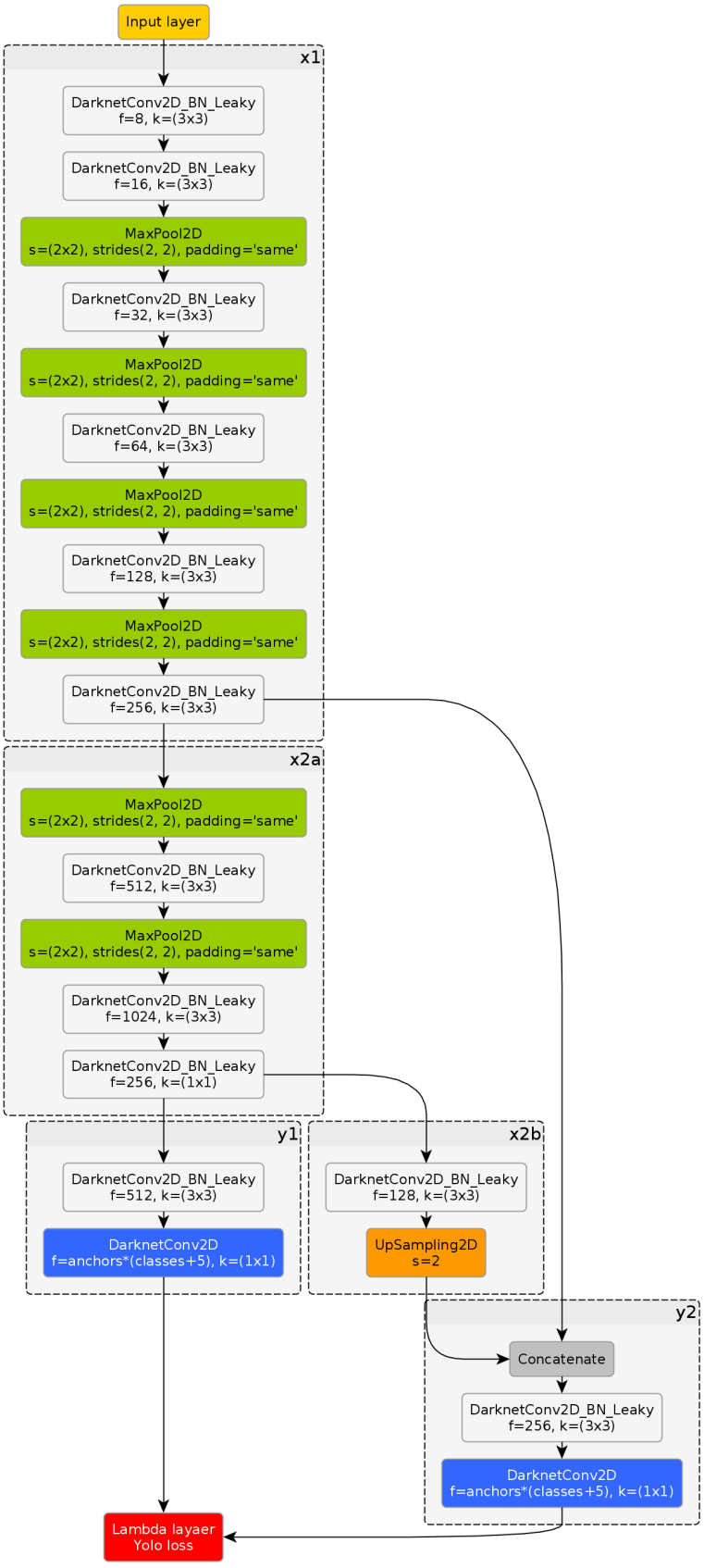

Figure 5-2: Architecture of the VVC1 network.

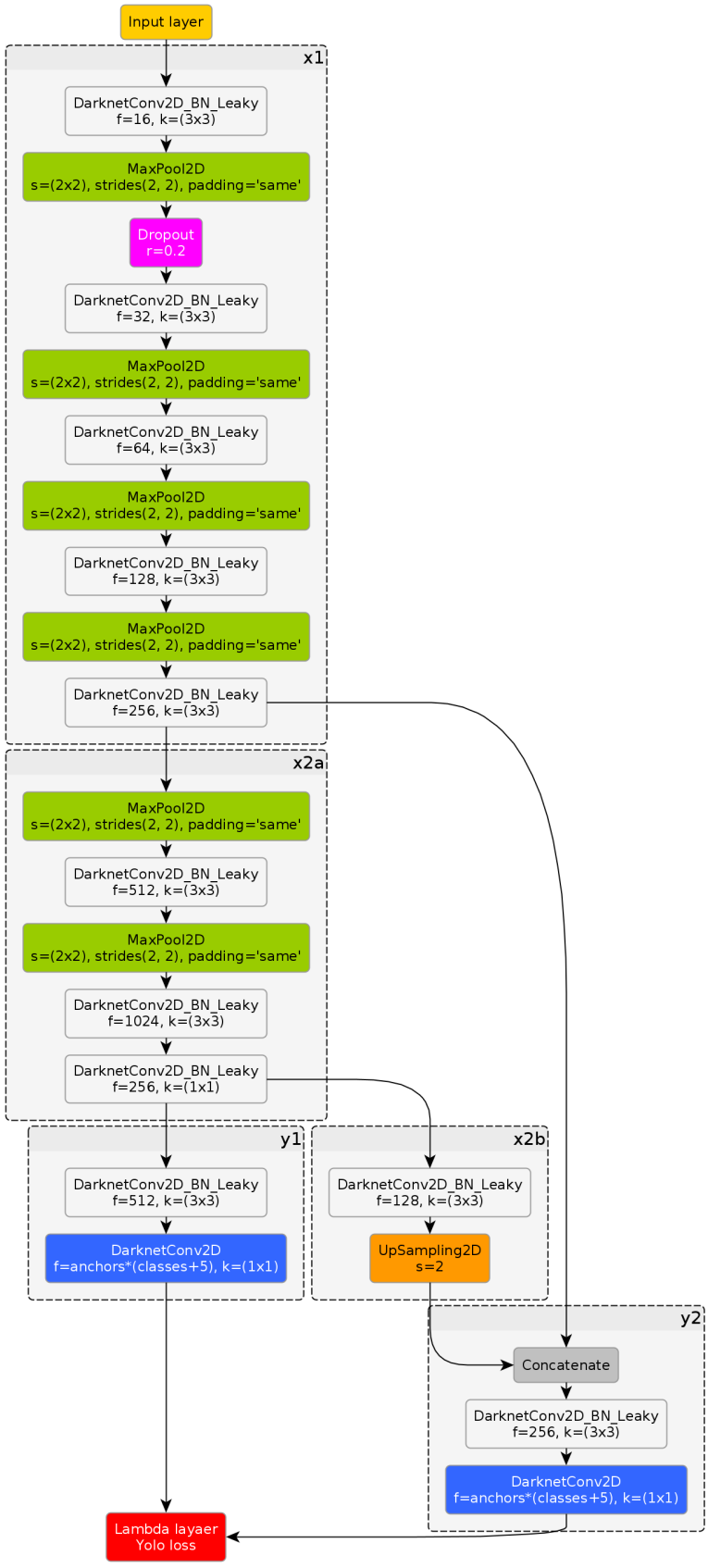

Figure 5-3: Architecture of the VVC2 network.

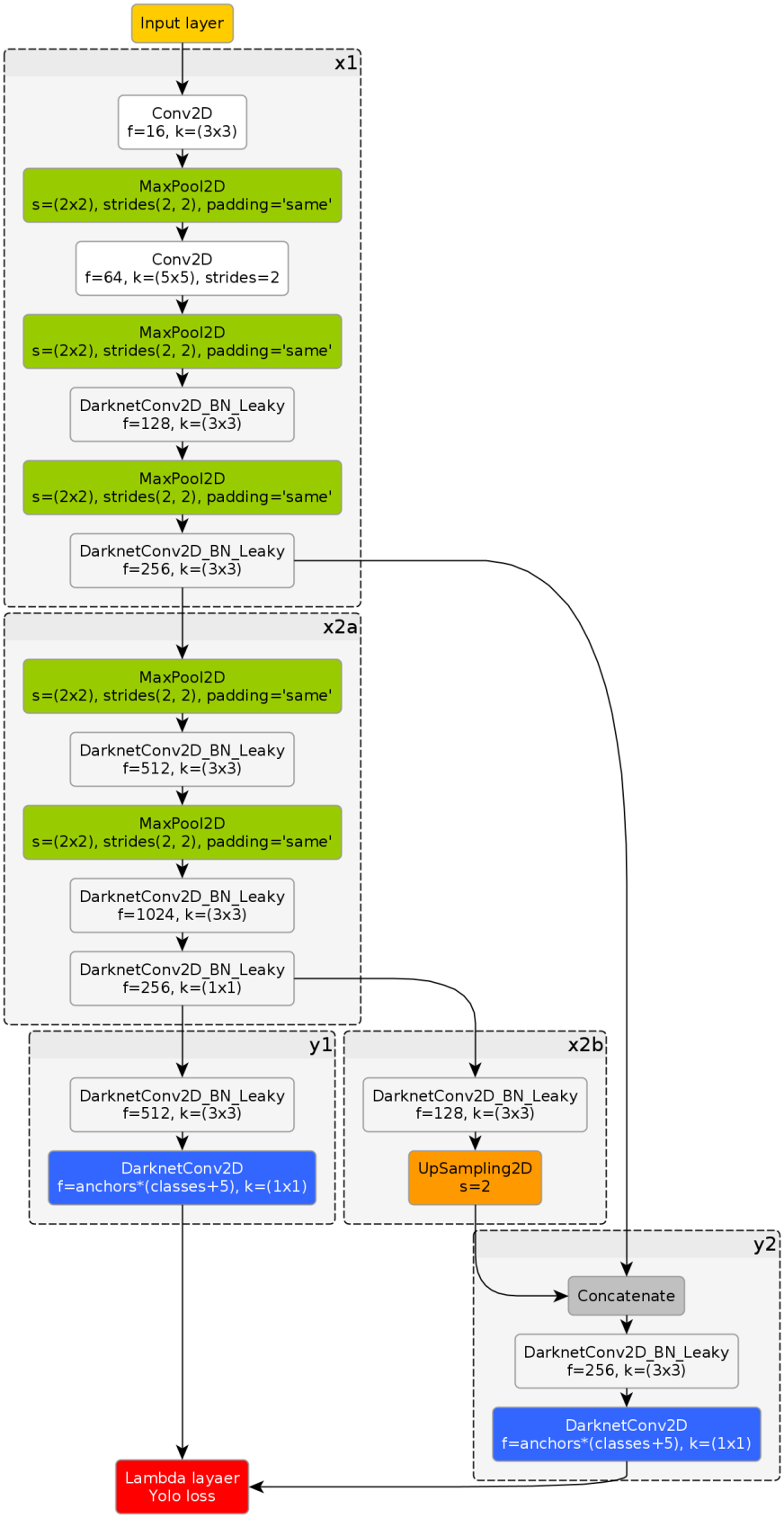

Figure 5-4: Architecture of the VVC3 network.

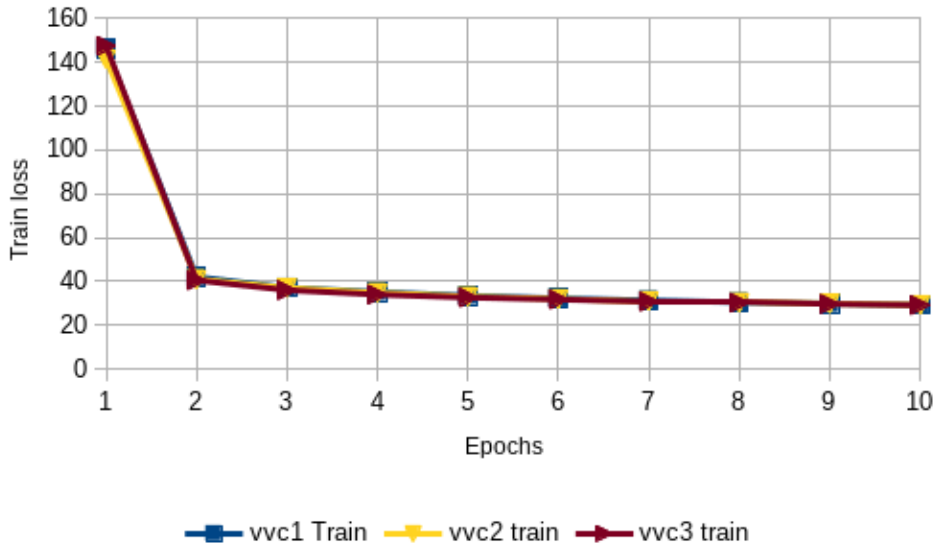

Figure 5-5: Training YOLO loss of the VVC networks with the COCOv dataset.

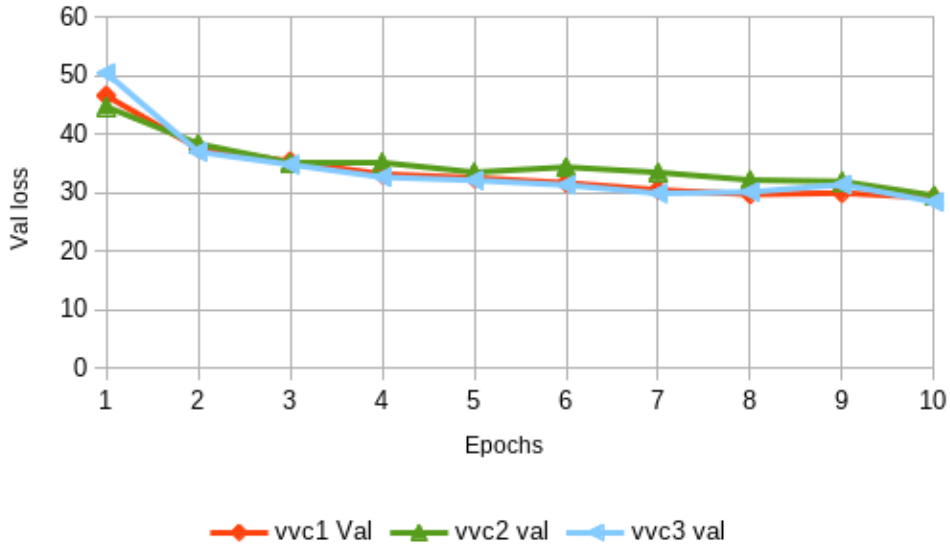

Figure 5-6: Validation YOLO loss of the VVC networks with the COCOv dataset.

6 System evaluation on surveillance videos

For the final evaluation of the counting system the Bog19 Dataset was used. The dataset was labeled with the CVAT tool introduced in [43] annotation tool. The ground truth is in a xml file with the tracks for each vehicle, including passengers in the case of motorbikes, the bounding boxes and a flag for occluded objects. The videos were captured with a smarphone camera at 30fps with a 1920x1080 resolution. Table **6-1** describes the dataset.

Table 6-1: Sumary of the Bog19 dataset.

Videos	Frames	Object boxes	Bicycles	Buses	Cars	Motorbikes	Trucks
2	2037	10196	3	2	32	11	1

The experiments run a desktop computer with a high end GPU for domestic use. The used GPU is the Nvidia Geforce 1080Ti with 11GB of VRAM, a CPU Ryzen 5 3600 and 16GB of RAM. The computer has Debian 10 GNU/Linux as operative system, with the 430.64 version of the Nvidia driver installed.

6.1. Tracking metrics

Eight metrics were chosen to evaluate the performance: Mostly Tracked targets (MT), Mostly Lost targets (ML), False Positives (FP), False Negatives (FN), Identity switches (IDs), Fragmentations (FM), Multi-Object Tracking Accuracy (MOTA) and Multi-Object Tracking Precision (MOTP).

In the figure **6-1**, the best MOTA value is for the VVC models with the Medianflow or MOSSE tracker. The Multiple Object Tracking Accuracy (MOTA) integrates the results for FN, FP, and IDs.

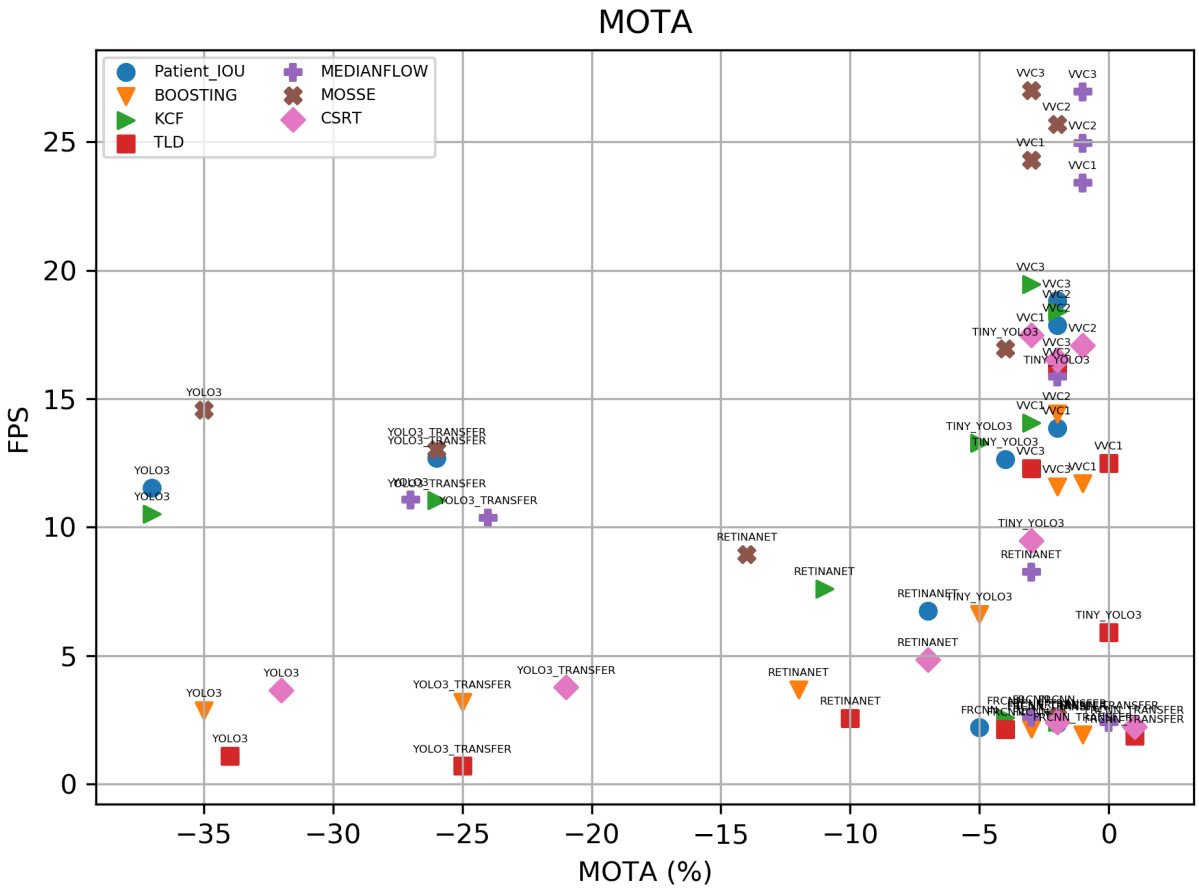

Figure 6-1: MOTA vs fps on the Bog19 dataset.

The MOTP metrics reflects the tracking precision based on the TP metric. In the figure **6-2**, the VVC models have the highest MOTP by fps. The RetinaNet networks achieve high MOTP but with more time cost.

The MT metric represents the number of correctly tracked ground truth trajectories in at least 80 % of the frames. In the figure **6-3**, the MT is divided by the number of ground truth trajectories, no combination of detector and tracker has a good MT value. This result indicates that the vehicles are track for a short periods of time.

The ML metric represents the number of correctly tracked ground truth trajectories in less than 20 % of the frames. In the figure **6-4**, the slower combinations track some vehicles, instead the faster ones lost many vehicle trajectories.

The IDs metric represents the vehicle is tracked but an incorrect ID is assigned to the trajectory. In the figure **6-5**, the VVC detectors with the Medianflow or MOSSE trackers

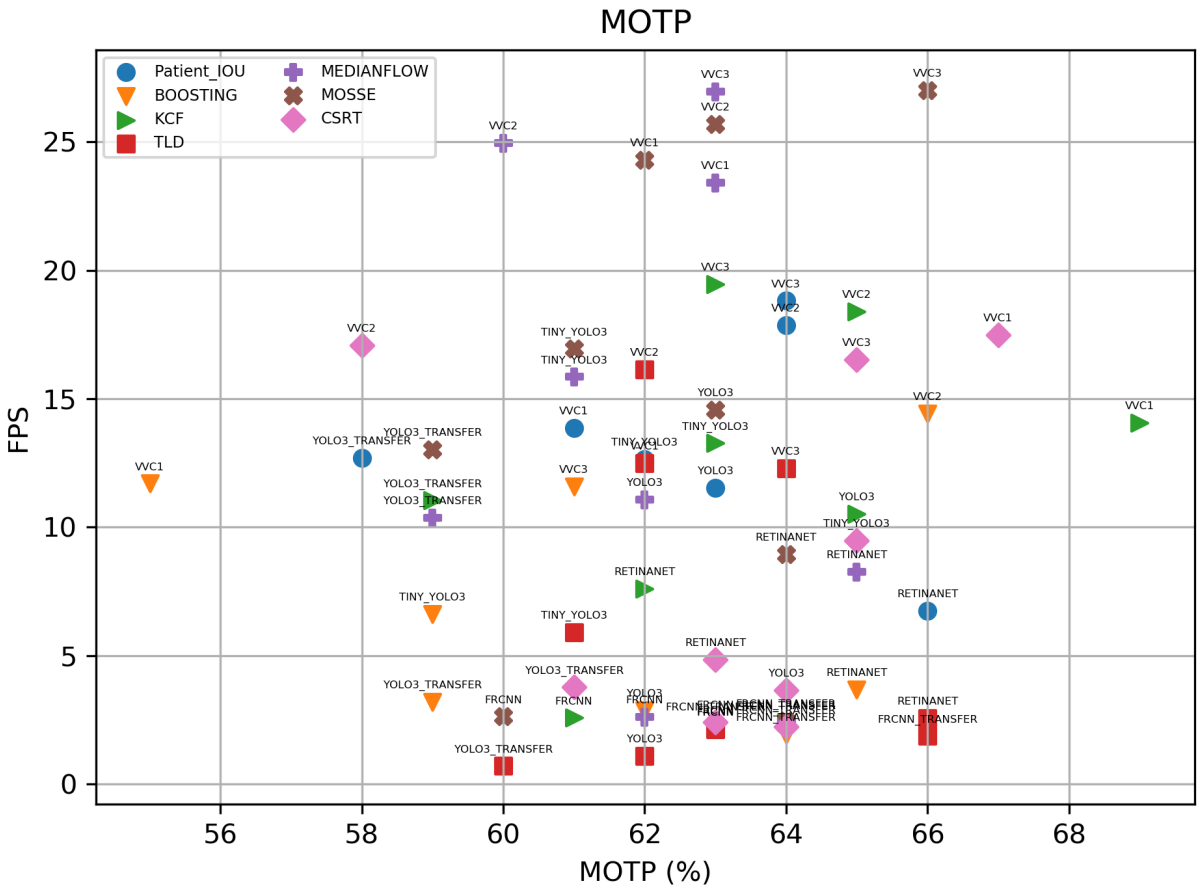

Figure 6-2: MOTP vs fps on the Bog19 dataset.

assign the correct trajectory ID with a speed higher than 20 fps.

The FM metrics represents the number of times a ground truth trajectory is interrupted and resumed. In the figure **6-6**, the Tiny YOLO and VVC networks has the lower fragmentations, but the VVC ones with the Medianflow or MOSSE trackers has the best speed.

The FN metric represents the number of ground truth bounding boxes no associate with a hypothesis bounding boxes. In the figure the **6-7**, most of the combinations fail to find the same group of bounding boxes.

The FP metric represents the number of hypothesis bounding boxes no associate with a real bounding boxes. In the figure **6-8**, most of the combinations can retrieve real bounding boxes. In particular the VVC detectors with Meandflow or MOSSE tracker achieve a good value at high speed.

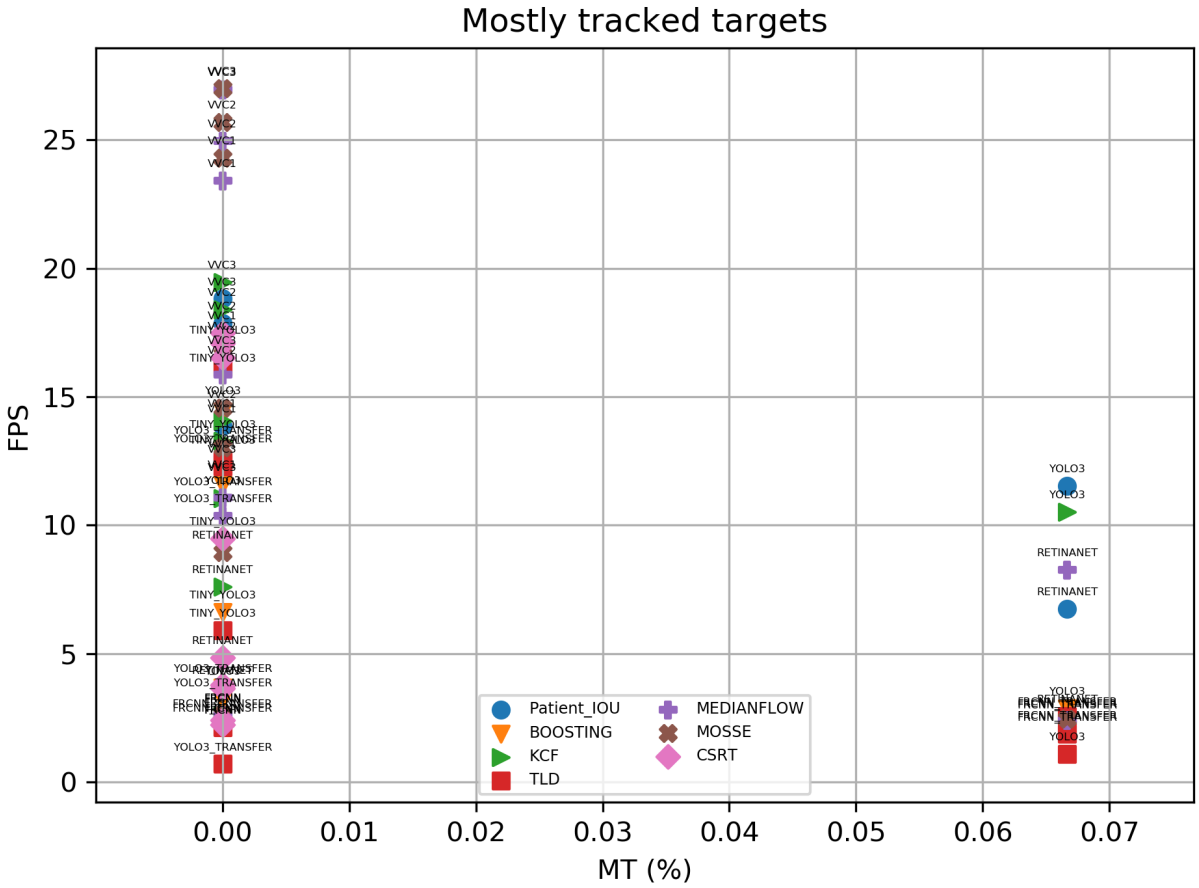

Figure 6-3: Mostly tracked objects (MT) vs fps on the Bog19 dataset.

6.2. Vehicle counting

A comparison of the system using different deep learning object detectors is made. The Faster R-CNN architecture with the resnet50 model, YOLOv3 and Tiny YOLO, and a version with transfer learning for each one, are compared keeping the same other phase configuration.

The videos used for training and evaluation are recordings of 3 of the main streets of Bogota, and had a resolution of 1090x1080. They were taken during the day from cameras installed over the streets at 30 fps. A manual process of tagging was made for the bicycle, car, motorbike, bus, and truck classes. The annotations include the visible faces of the vehicle and a flag for object occlusion.

The counting precision is defined as the average of correct identified vehicles on each frame. The absolute difference between the expected and predicted counts by class over the expected

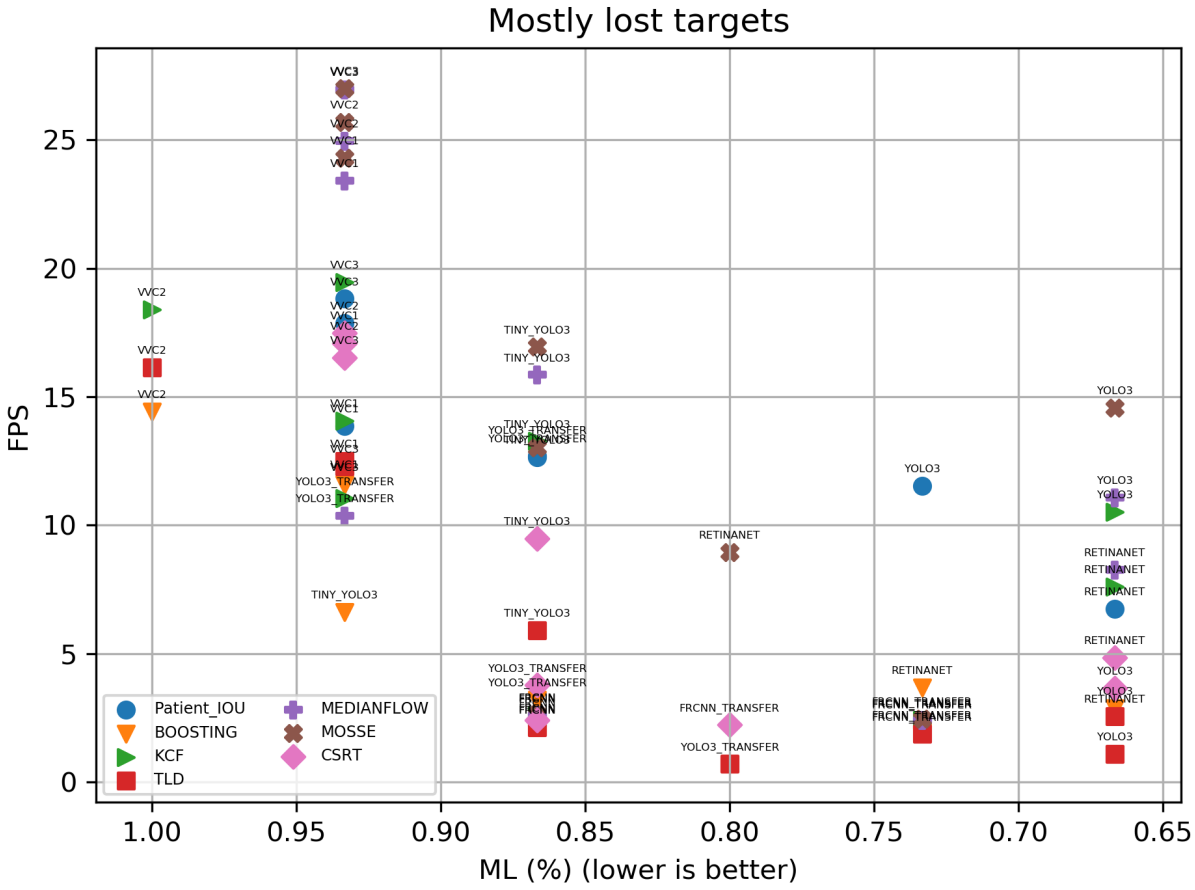

Figure 6-4: Mostly lost targets (ML) vs fps on the Bog19 dataset.

is the measurement used. The figure **6-9** shows that the pre-trained models as Faster R-CNN and YOLOv3 have the higher average counting precision for buses and motorbikes.

Of the 4 phases, the detection one consumes most of the time. In the figure **6-11** the average frame time of each phase is plotted. The tracking phase has almost zero time because the simple tracking algorithm only uses the bounding boxes of the previous phase and make no image processing.

The figure **6-12** shows that at 1066x600 resolution Tiny YOLO with transfer learning is the fastest one. The fps are highly dependent on the input resolution and speed of the detector. The re-trained models are faster than the pre-trained versions, because the small number of classes to predict.

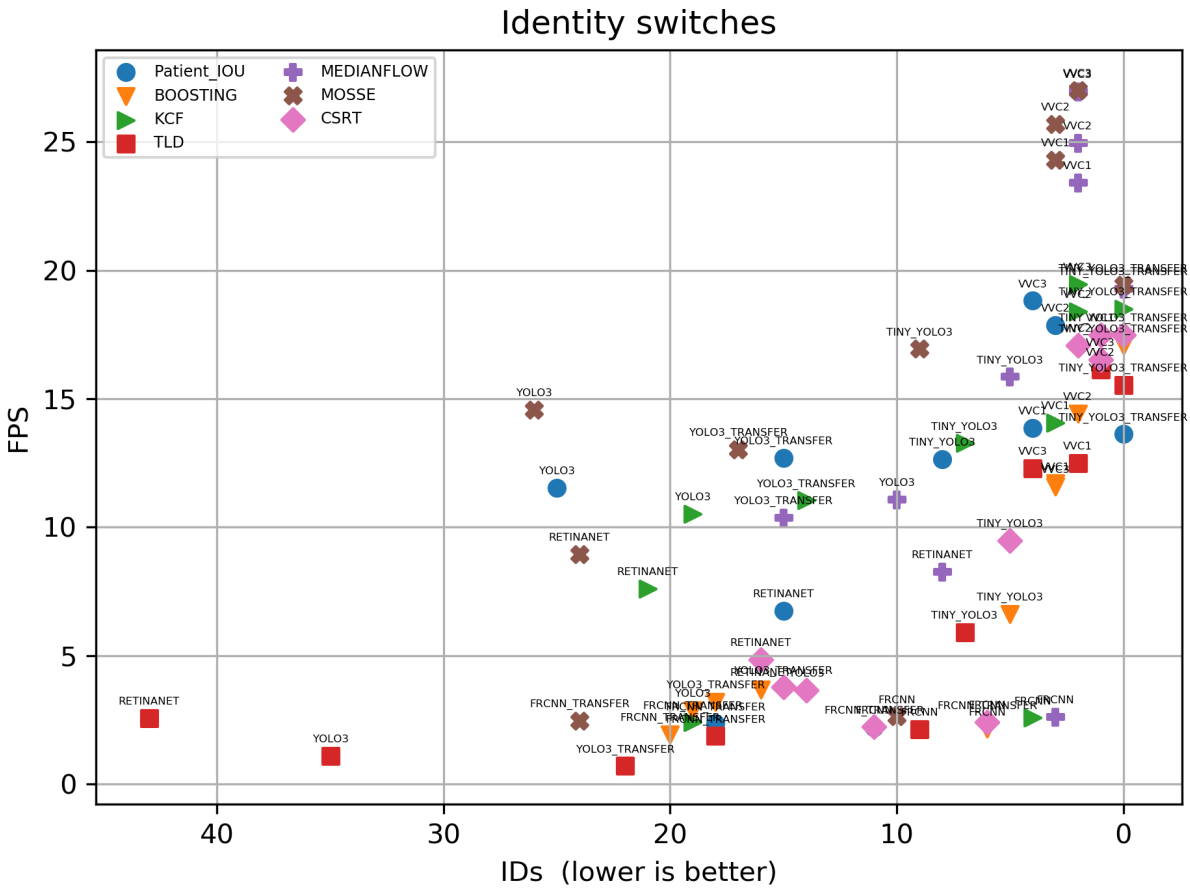

Figure 6-5: Identity switches vs fps on the Bog19 dataset.

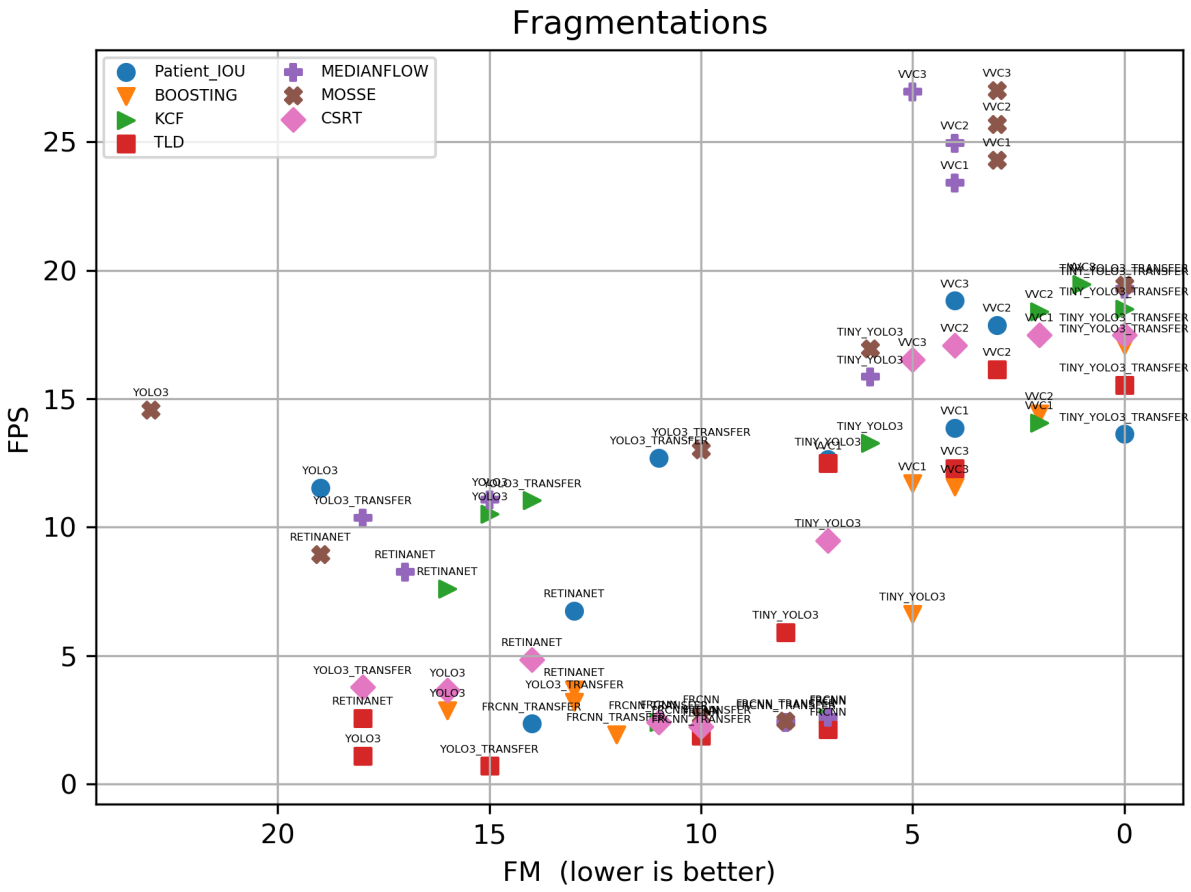

Figure 6-6: Fragmentations vs fps on the Bog19 dataset.

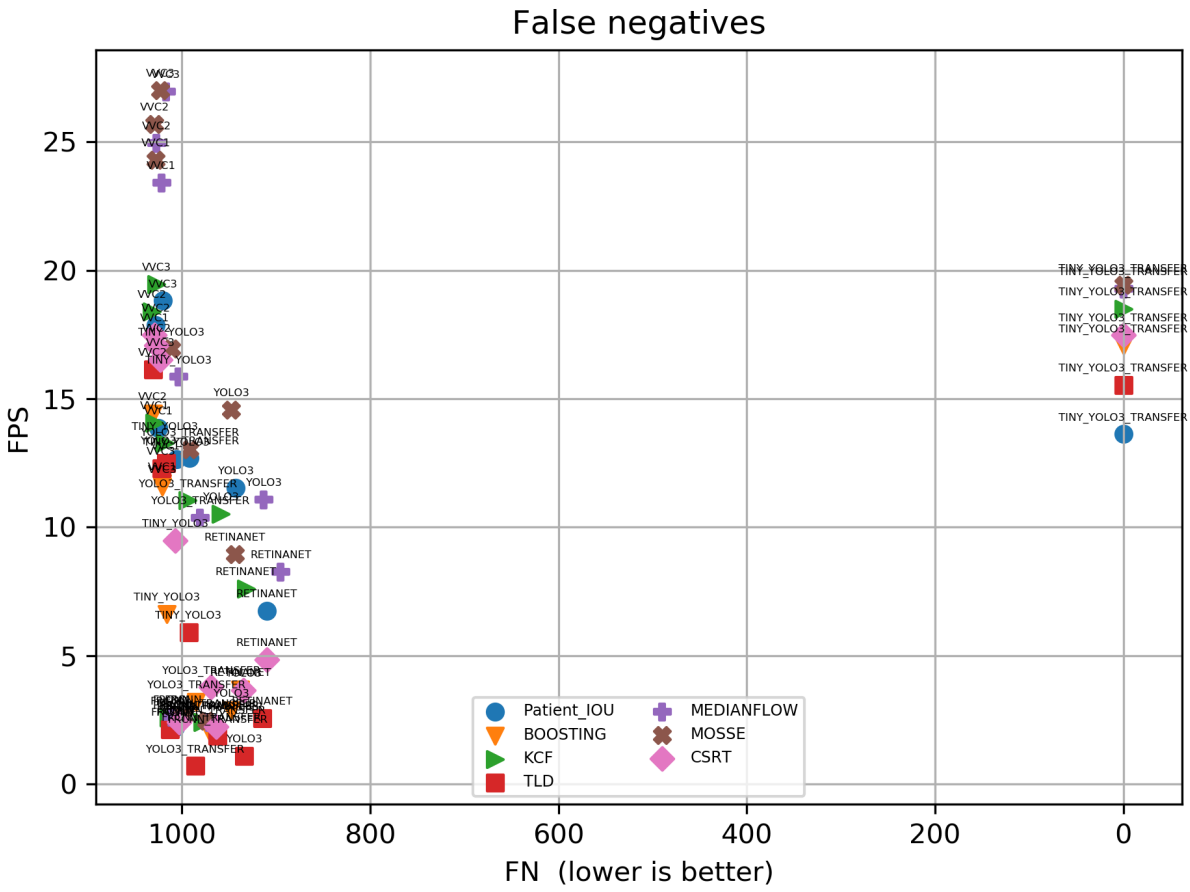

Figure 6-7: False negatives vs fps on the Bog19 dataset.

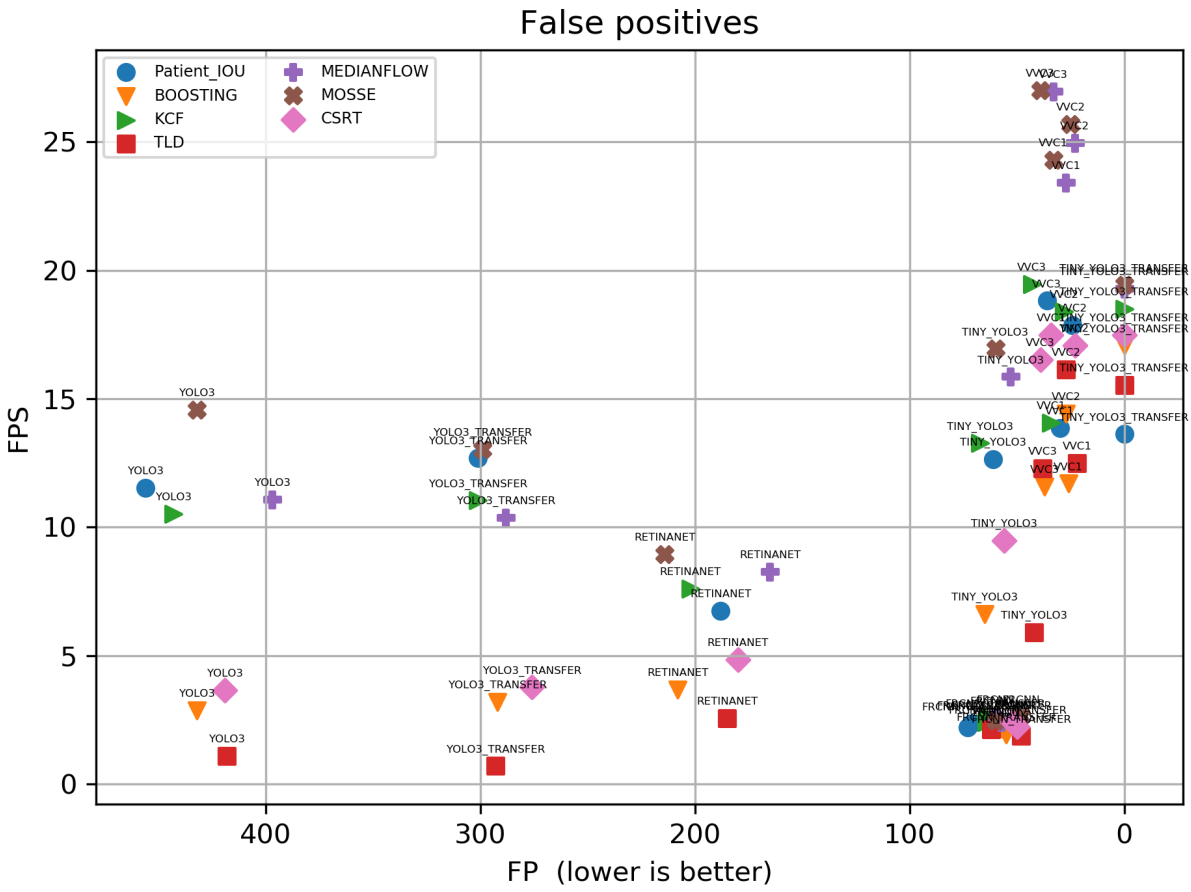

Figure 6-8: False positives vs fps on the Bog19 dataset.

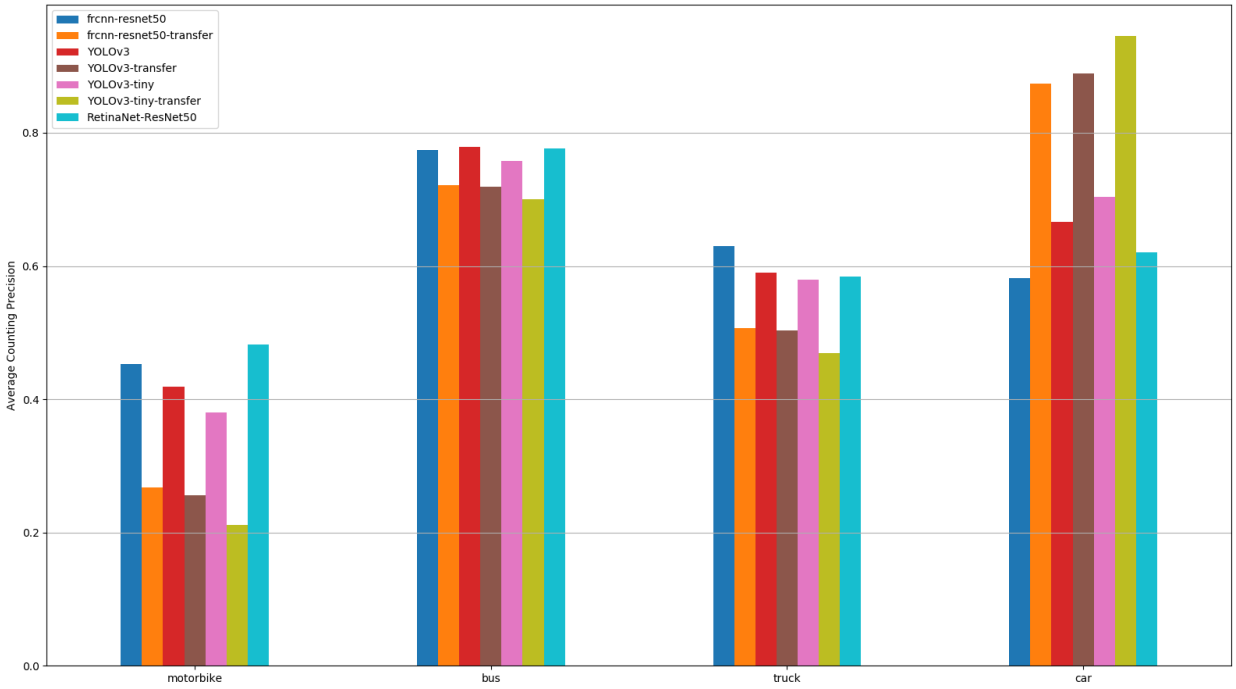

Figure 6-9: Average counting precision.

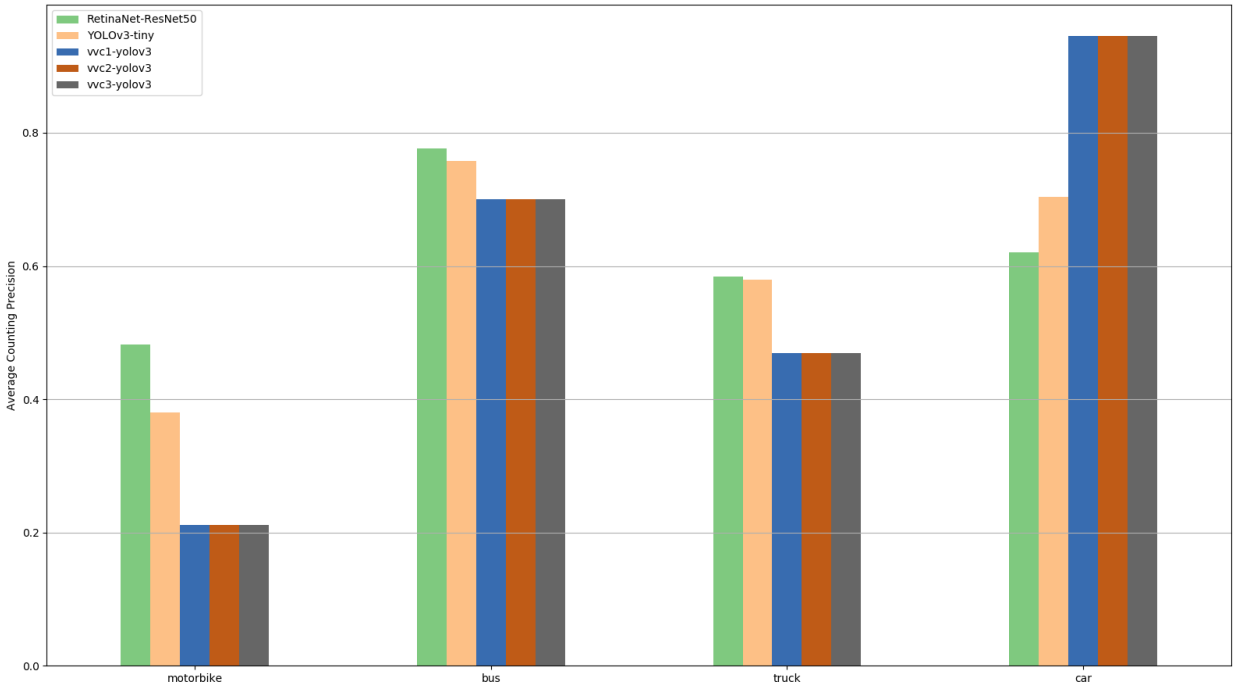

Figure 6-10: Average counting precision.

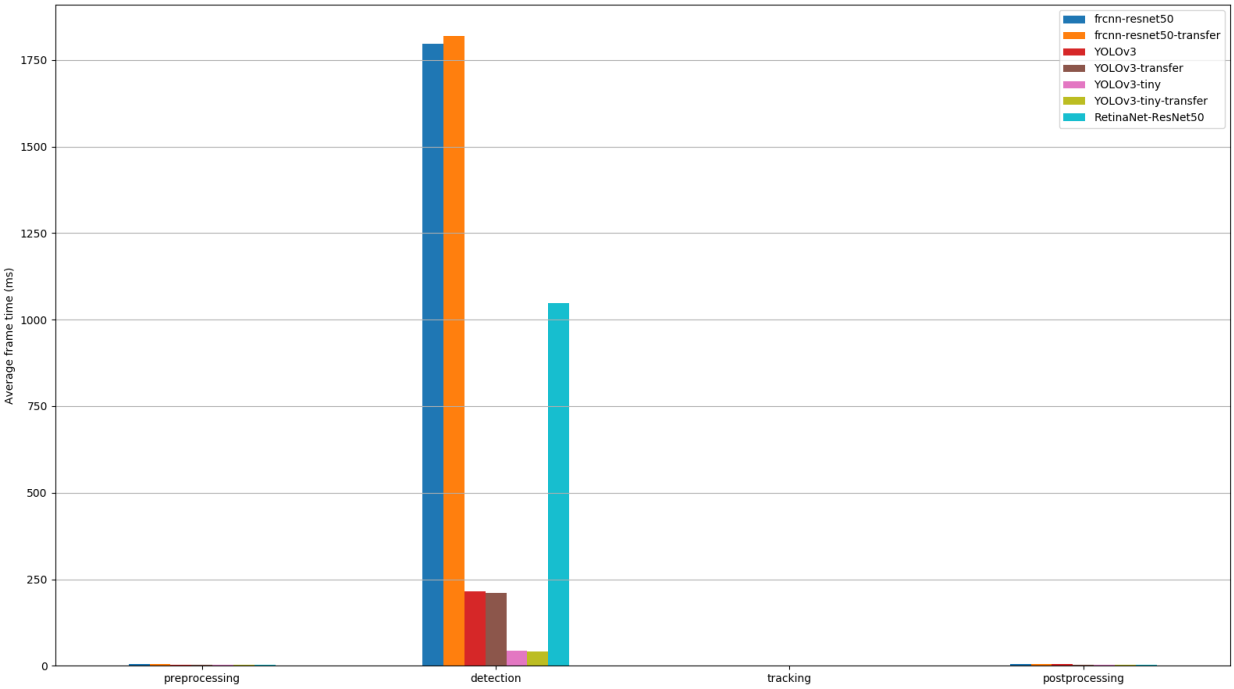

Figure 6-11: Average frame time by phase.

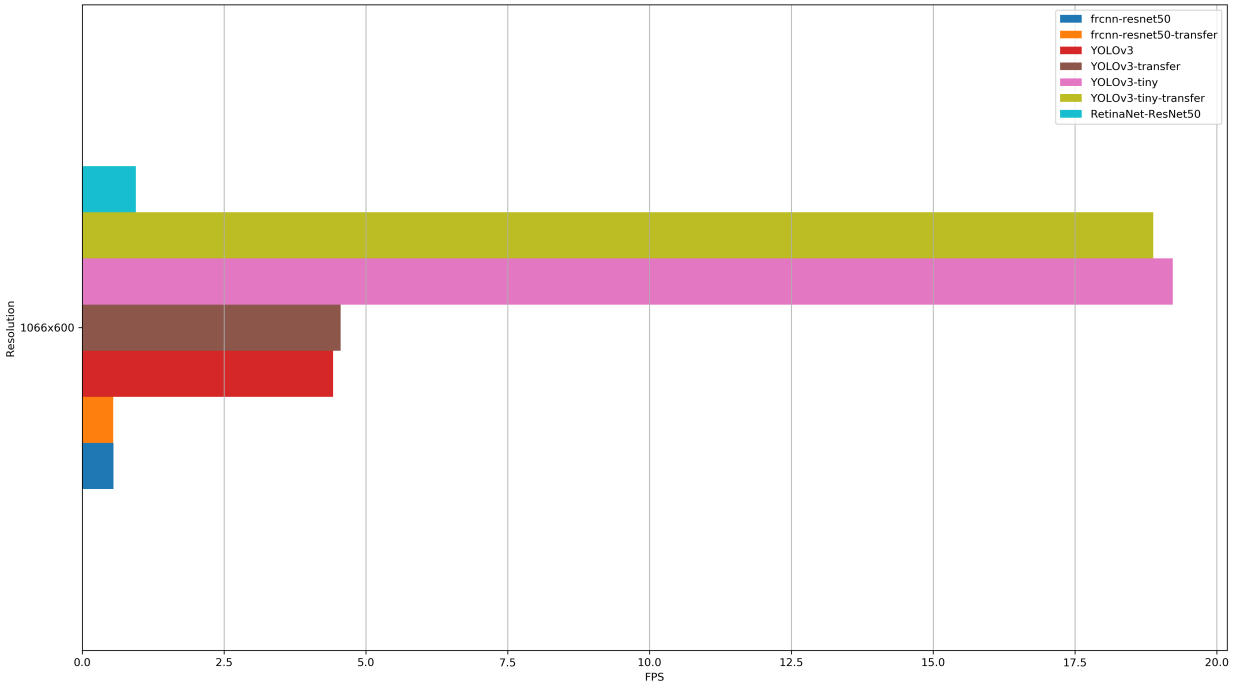

Figure 6-12: Average FPS using different detectors.

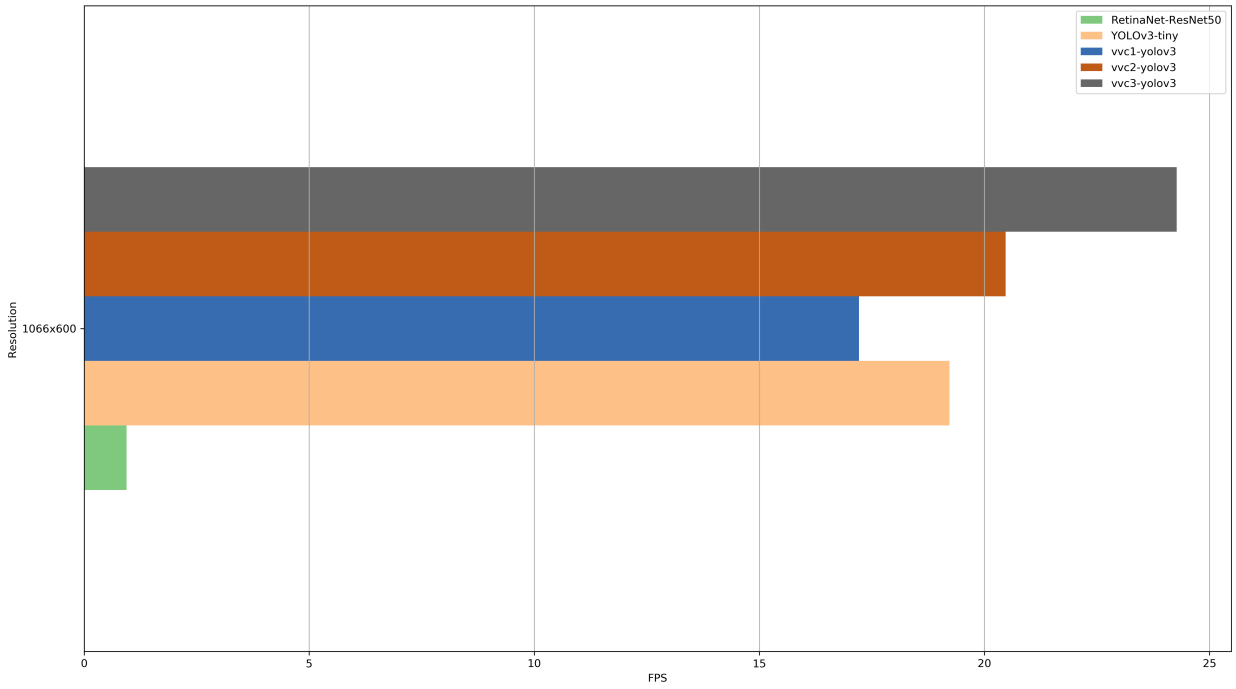

Figure 6-13: Average FPS using different detectors.

7 Conclusions and recommendations

7.1. Conclusions

In this work, we presented the desired properties of a robust object detection method, 3 new vehicle detectors, 1 new vehicle tracker, the development of a system for visual counting, and the performance of the system in a custom dataset. The following characteristics of a robust detection method were described. tolerance to variable scenes, camera location, camera movement, image color, image resolution and the most challenging, object occlusion.

The VVC detectors are CNN based on Tiny YOLOv3, with modifications on the first layers of the original network. The modifications include aggregation, removal and alteration of the layer hyper-parameters. The Patient IOU tracker is an extension of the IOU tracker introduced by Bochinski [36]. The new tracker can reactivate the tracks if a detection match the last seen bounding box of the trajectory.

A software system was developed to integrate known and new detectors and trackers. The evaluation of the system is based on the MOT metrics using the Bog19 dataset. The Bog19 dataset is a collection of annotated videos obtained from the traffic authority of Bogotá city. The best performing combination is the VVC3 network with the Medianflow tracker, followed by the same network with MOSSE tracker.

The counting performance is highly dependent on the quality of the findings of the vehicle detection phase. If the vehicles are not detected, then there is no effective tracking and counting. The counting results with the VVC detectors are similar to detectors of the state of the art but with higher speed.

The detection phase consumes most of the processing time. Consequently, a faster algorithm for detection allows the use of the system for real time jobs. The information obtained can be disseminated to road users, potentially reducing congestion and improving traffic safety. For example, traffic density on major roads can be estimated and less congested routes and shorter travel times can be calculated and transmitted to drivers.

The counting results can be used as input to traffic models and urban planning process for the Bogotá city. Using automatic systems for vehicle detection, tracking and traffic analysis would be very useful for the city's ITS. The system reduces costs by not having to install new sensors and, improves the frecuency and processing time of the current manual couting. The information of the project and source code are available online [1].

7.2. Recommendations

The following recommendations for future work arise from the present research:

- Use and compare the SSD object detector and others for the task of vehicle detection to increase the spectrum of detectors for the system.
- Increase the available training dataset and time to improve the detection performance of the VVC networks.
- Search for Automated Machine Learning (AutoML) techniques to automatically find specialized networks and build pipelines from training to deployment.
- Use tracking algorithms based on deep networks and evaluate the possibility of using a single network for detection and tracking of vehicles.
- Evaluate the system capacity to operate with the input from multiple surveillance cameras at the same time.

[1] https://vvc-unal.github.io

Bibliography

[1] N. Buch, S. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011. cited By 171.

[2] "Citilog," Nov. 2016.

[3] E. TIEMPO, "Bogotá tendrá nuevo centro de control para vigilar el tráfico," *EL TIEMPO*, Dec. 2015.

[4] K.-S. Jie and M. Liu, "Computer vision based real-time information acquisition for transport traffic," in *ICIA 2005 - Proceedings of 2005 International Conference on Information Acquisition*, vol. 2005, (Hong Kong), pp. 164–169, 2005.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 2, pp. 1097–1105, 2012.

[6] Y. Liu, B. Tian, S. Chen, F. Zhu, and K. Wang, "A survey of vision-based vehicle detection and tracking techniques in ITS," in *Proceedings of 2013 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2013*, pp. 72–77, 2013. cited By 3.

[7] E. Baş, A. b. b. Tekalp, and F. Salman, "Automatic vehicle counting from video for traffic flow analysis," in *IEEE Intelligent Vehicles Symposium, Proceedings*, (Istanbul), pp. 392–397, 2007.

[8] L. Li, W. Huang, I. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, (Berkeley, CA.), pp. 2–10, 2003.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*,

vol. 2017-October, pp. 2999–3007, Institute of Electrical and Electronics Engineers Inc., 2017.

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, vol. 1, pp. I–511–I–518 vol.1, 2001.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005.

[13] P. Druzhkov and V. Kustikova, "A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognition and Image Analysis*, vol. 26, no. 1, pp. 9–15, 2016.

[14] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 11-18-December-2015, pp. 1440–1448, Institute of Electrical and Electronics Engineers Inc., 2015. cited By 23; Conference of 15th IEEE International Conference on Computer Vision, ICCV 2015 ; Conference Date: 11 December 2015 Through 18 December 2015; Conference Code:119541.

[15] S. b. 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* (G. R. L. N. S. M. Cortes C., Lee D.D., ed.), vol. 2015-January, pp. 91–99, Neural information processing systems foundation, 2015. cited By 75; Conference of 29th Annual Conference on Neural Information Processing Systems, NIPS 2015 ; Conference Date: 7 December 2015 Through 12 December 2015; Conference Code:120037.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, "SSD: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.

[17] A. Wong, M. Shafiee, F. Li, and B. Chwyl, "Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection," in *Proceedings - 2018 15th Conference on Computer and Robot Vision, CRV 2018*, pp. 95–101, 2018.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-January, pp. 779–788, 2016.

[19] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6517–6525, 2017.

[20] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.02767, 2018.

[21] Q. Fan, L. Brown, and J. Smith, "A closer look at Faster R-CNN for vehicle detection," in *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2016-August, pp. 124–129, Institute of Electrical and Electronics Engineers Inc., 2016. cited By 0; Conference of 2016 IEEE Intelligent Vehicles Symposium, IV 2016 ; Conference Date: 19 June 2016 Through 22 June 2016; Conference Code:123227.

[22] Wahyono, V. D. Hoang, L. Kurnianggoro, and K. H. Jo, "Scalable histogram of oriented gradients for multi-size car detection," in *2014 10th France-Japan/ 8th Europe-Asia Congress on Mecatronics (MECATRONICS)*, pp. 228–231, Nov. 2014.

[23] B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (Toronto, ON), pp. 1078–1083, 2006. cited By 28; Conference of ITSC 2006: 2006 IEEE Intelligent Transportation Systems Conference ; Conference Date: 17 September 2006 Through 20 September 2006; Conference Code:71696.

[24] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, Mar. 1960.

[25] T. Mauthner, M. Donoser, and H. Bischof, "Robust tracking of spatial related components," in *Proceedings - International Conference on Pattern Recognition*, (Tampa, FL), 2008. cited By 8; Conference of 2008 19th International Conference on Pattern Recognition, ICPR 2008 ; Conference Date: 8 December 2008 Through 11 December 2008; Conference Code:81859.

[26] F. Bardet and T. Chateau, "MCMC particle filter for real-time visual tracking of vehicles," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (Beijing), pp. 539–544, 2008.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy,

A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[29] 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," *Lecture Notes in Computer Science*, pp. 740–755, 2014.

[30] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking," *arXiv CoRR*, vol. abs/1511.04136, 2015.

[31] P. Zhu, L. Wen, X. Bian, L. Haibin, and Q. Hu, "Vision Meets Drones: A Challenge," *arXiv preprint arXiv:1804.07437*, 2018.

[32] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: Multi-Object Tracking and Segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] Y. c. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[34] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[35] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[36] E. Bochinski, V. Eiselein, and T. Sikora, "High-Speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*, 2017.

[37] E. Bochinski, T. Senst, and T. Sikora, "Extending IOU Based Multi-Object Tracking by Visual Information," in *Proceedings of AVSS 2018 - 2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, 2019.

[38] E. E. Hilbert, P. A. Rennie, and W. A. Kneidl, "A sensor for control of arterials and networks," *IEEE Transactions on Vehicular Technology*, vol. 29, pp. 208–215, May 1980.

[39] J.-W. b. Hsieh, S.-H. b. Yu, Y.-S. b. Chen, and W.-F. b. Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 179–186, 2006. cited By 169.

[40] Y. b. Lecun, Y. Bengio, and G. e. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[41] F. Valcárcel, C. Chaves, R. Moreno, and O. Sanchez, "Machine vision algorithms applied to dynamic traffic light control [Algoritmos de visión de máquina aplicados al control dinamico de intersecciones semáforizadas]," *DYNA (Colombia)*, vol. 80, no. 178, pp. 132–140, 2013. cited By 0.

[42] C. Pang, W. Lam, and N. Yung, "A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 441–459, 2007.

[43] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," *International journal of computer vision*, vol. 101, no. 1, pp. 184–204, 2013.