Robust Parking Block Segmentation from a Surveillance Camera Perspective

Nisim Hurst-Tarrab[a,*], Leonardo Chang[a], Miguel González-Mendoza[a]

[a] *Tecnologico de Monterrey, Escuela de Ingenieria y Ciencias, Mexico*

Abstract

Parking block regions host dangerous behaviors that can be detected from a surveillance camera perspective. However, these regions are often occluded, subject to ground bumpiness or slope and thus they are hard to segment. Firstly, the paper proposes a pyramidal solution that takes advantage of satellite views of the same scene, based on a deep Convolutional Neural Network (CNN). Training a CNN from the surveillance camera perspective is rather impossible due to a combinatory explosion generated by the distinct point-of-views. However, CNNs showed great promise on previous works on satellite images. Secondly, even though there are many datasets like PKLot (Almeida et al. 2015) or CNRPark-EXT (Amato et al. 2017) for occupancy detection in parking lots, none of them were designed to tackle the parking block segmentation problem directly. Given the lack of a suitable dataset, we also propose APKLOT, a dataset of roughly 7000 polygons for segmenting parking blocks from the satellite perspective and from the camera perspective. Moreover, our method achieves more than 50% intersection over union in all the testing sets, i.e., at the satellite view and the camera view.

Keywords: deep learning, parking lot dataset, parking block segmentation, satellite dataset

1. Introduction

Parking lots are dynamic environments brew for mishaps of many sorts. A crime investigation report of the Bureau of Justice Statistics of the United States in parking garages facilities states that 9% of 2010 crimes occurred in parking places (Harrell 2012). Algorithms for simulating and controlling parking lot behavior e.g., vacant space detection, rely heavily on having parking spot and parking block areas previously marked by a human (Sevillano, Mármol, and Fernandez-Arguedas 2014).

In this work we propose to segment parking blocks in a surveillance camera image. The previously mentioned insecurity problem can then be ameliorated by a second generative algorithm (out of the scope of this work) that take benefit from the priors our proposal produces. This approach was first used for traffic scene analysis in (Huang et al. 1994), by using dynamic belief (Bayesian) networks. In the same vein, (Seo, Ratliff, and Urmson 2009) proposes to use prior probabilities of parking block areas of these spaces to guide robot navigation.

Seo also provides a definition for parking blocks in terms of parking spots. We extended this definition to include also partially occluded parking spots and of arbitrary shapes onto splines and containing parking spots of different sizes or neighborhood arrangements.

*Corresponding author

Email addresses: langheran@gmail.com (Nisim Hurst-Tarrab), lchang@tec.mx (Leonardo Chang), mgonza@tec.mx (Miguel González-Mendoza)

In brief, segmenting parking blocks means to be able to identify areas in the ground plane that belong to a series of parking blocks, automatically and with high resiliency over the particular surveillance camera view conditions. There are challenges derived by the problem formulation by itself.

Firstly, an obvious challenge is the different angles and zooms a camera could have. The combinations of the three components of the projection angle and any reasonable zoom level set by the surveillance operator generates an exponentially growing set of possible projection planes. Even though we can still apply an affine transformation to normalize into a simpler domain (Sastre et al. 2007), each pose yields a slightly different set of unrelated features. Considering these changes in the point of view, the most common features can become more or less visible.

Secondly, in the worst case scenario, each scene is particularly packed with a series of occlusions and shadows depending on the particular camera position. This phenomenon introduces a Bayesian noise that is impossible to overcome without lots of data samples.

Third, we also have a broader range of noise sources from the camera perspective in detecting parking spots. (Weis, May, and Schmidt 2006) categorize them into 3 main sources:

1. **Typical Attributes.** Similar painted lines, other vehicles, kerbstones, non-plain floor, etc.
2. **Road Conditions.** Partially damaged parking lines, paving, etc.
3. **Surrounding Conditions.** Weather, illumination, light source angle, etc.

We can add to this list the presence of occlusions caused by temporal entities like vehicles and pedestrians. Also (but more rare though), is the usage of cameras with different settings like color resolution, in the same parking lot.

For these three aforementioned reasons, solving our problem by a mere (self-)supervised algorithm would require an exponential number of samples, and thus is virtually impossible.

In the past 25 years approaches to solve this problem obviated the use of satellite perspective images available for the same surveillance camera scene. With the advent of modern public Geographic Information Systems (GIS) repositories, high resolution satellite perspective images of outdoor city components became ubiquitous. These satellite perspective images frequently include parking lots, and in some cases, they are even segmented by community members. The satellite perspective allows the following advantages:

- Parking lots are viewed from the space.
- Texture appearance prevails over other features.
- The view is orthogonal.
- Instances can be rotated and form a manifold with the sole condition they do not overlap.
- Variable angle relative to the road, no more than 180 degrees.
- Satellite images rarely have shadows and we could remove them by using histogram normalization.

Furthermore, (Seo, Ratliff, and Urmson 2009) demonstrated that canonical parking spots are easier to detect in those images. Also, (Kabak and Turgut 2010) provides a good summary of features in the satellite realm.

To take advantage of these previous findings, we propose to segment the parking blocks of the surveillance camera image by:

1. Segment the parking blocks on the satellite image.
2. Calculate an homography between the two perspectives.
3. Translate the results of the satellite image into the surveillance camera image.

The beauty of this approach is that it helps us overcome the two main challenges previously mentioned: (1) temporal and static frontal[1] occlusions and (2) variations of the point of view that thwarts a supervised training approach. See Figure 1 for an illustration of our proposal.

Nowadays available datasets are not enough for training a neural network, further on we explain why is this so in the datasets section. Nonetheless, we saw this challenge as a great opportunity to make a contribution to the computer vision field for identifying human made structures from the sky. For the previously mentioned reason -and in particular- to be able to train our image segmentation model in the satellite perspective, we introduce *APKLOT*, a dataset for direct parking block segmentation from a satellite perspective. More details about APKLOT[2] dataset will be given in Section 3.

Our image segmentation model was inspired by the publicly available convolutional neural network for image segmentation implementation in dlib (King 2009) with custom parametrization. For the homography calculation step we used a custom python tool that helped us mark the correspondence points, even though this wouldn't be necessary if we had had the physical setup camera projection parameters. These steps will be explained in further detail in the following sections.

Finally, we also propose an evaluation protocol based on the work of . Companion code for all the aforementioned tasks is also provided in the github repository.p

The paper is organized as follows. Section 1.1 explains why this problem is important in the context of the current available related work and the specific way in which we will measure success. Section 2 describes an overview of the method, making emphasis on the implemented convolutional neural network and from which related works does our intuition stems. Section 3 presents the dataset we used to train our unique approach to parking block segmentation. Section 4 proposes a set of experiments to evaluate the dataset in the face of the most influential hyperparameters proposed in the literature. It also places each result along the method step's, thus providing a clearer perspective of their contribution. Section 5 remarks the paper's most valuable contributions and detours into further developments that can be supported by our work.

1.1. Related Works

Most prior work focus on detecting a global parking lot structure. Of the most prominent examples stands out Huang et al. with a series of works (Huang et al. 2008; Huang and Wang 2010) that calculate global parameters such as angle to the traffic lane and distance between each parking spot per parking lot.

The works of (Huang and Wang 2010; Mexas and Marengoni 2014) and more recently (Koutaki, Minamoto, and Uchimura 2016); all use a two tier combination of vehicle detection with free (vacant) parking spot detection combination to determine structure global parameter in the parking blocks.

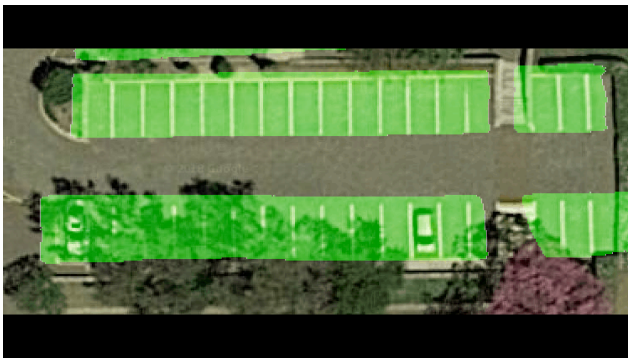
Free parking spot detection and segmentation usually use human-painted parking lot demarcations. These are ground appearance features that mostly lie on the same plane and are better visible from a high resolution satellite view (Cheng et al. 2014). They have been exploited extensively through the literature (Jung et al. 2006; Mexas and

[1] It is worth noting that **overhead occlusions** can be an emerging problem but they are compensated by a clearer shot of the overall parking block structure as we will see later.

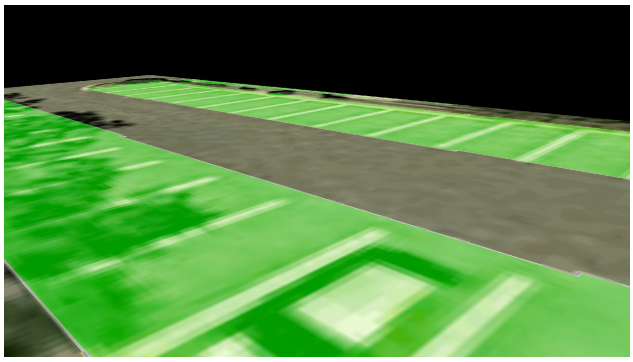
[2] *APKLOT* dataset files are available on the author's Github account (http://github.com/langheran/APKLOT). **Current version**: May 06, 2019.

(a) input

(b) satellite segmentation



(c) seg.+homography



(d) final result

Figure 1: Segmented parking blocks despite several frequent vehicle occlusions from a surveillance camera perspective.

Marengoni 2014), for example using the Hough transform. Other features can be extracted from movement in video (Sevillano, Mármol, and Fernandez-Arguedas 2014) and from the spatial distribution of known entities like other cars (Hsieh, Lin, and Hsu 2017).

Our work address the more general problem of extracting the parking lot structure from a single satellite image *without making any global shape assumptions*. In this way, we are able to detect parking blocks of arbitrary shapes.

Also, most of prior work focus on first detecting parking spots and then assembling those instances into a set characterized by a single partition function i.e., a parking block (Huang et al. 2008; Seo, Ratliff, and Urmson 2009). On these settings, a considerable chunk of the global information encoded in the image is then dismissed. This information is crucial for detecting partially occluded parking spots.

In contrast, our work doesn't waste any global information because it takes the pyramidal approach and starts by detecting parking blocks. In this work we do not attempt to detect individual parking spots. However, given the Occam's razor principle and Joshua Tenenbaum's **size principle**[3] (Murphy 2013) a smaller hypothesis localized only on a single parking block would make global assumptions more consistent with the data.

[3]Also known as the **Occam's razor** principle inspired in the English monk and philosopher William of Occam.

2. Camera Perspective Parking Block Segmentation Method

We previously mentioned how satellite views simplify the task of recognizing parking spots. These approaches have been neatly fitted in a two-category taxonomy by Huang (Huang and Wang 2010): car-driven and space-driven. However, in this work we took a joint approach that detects parking blocks directly.

CNNs were successfully used on detect parking lot patches (Cisek et al. 2017) with best test classification accuracy of 94.3%. (Amato et al. 2017) uses a CNN based on AlexNet for occupancy detection over *PKLot* and their own dataset *CNRPark*, achieving roughly a 3% overall accuracy improvement compared with previous methods based on two textural descriptors, namely Local Binary Patterns and Local Phase Quantization presented in (Almeida et al. 2015). The Cityscapes challenge (Cordts et al. 2016) provides segmented parking areas (not necessarily parking blocks). However, Cityscapes does not provide accuracy values for evaluating this class performance and whenever a car or other object is in front, the resulting class is this other object and not the parking area.

One domain from a satellite perspective that is remarkably similar to ***parking*** *block segmentation* is ***building*** *block segmentation* (Marmanis et al. 2016, 2018; Demir et al. 2018). In fact, the frontpage image of (Marmanis et al. 2016) shows parking block areas segmented by mistake.

This is a clear hint that their method would perform well also on parking blocks, given that our method could be considered a more general case. The best results in both cases were obtained by fully convolutional networks (FCNs) with 88.5% overall accuracy and 0.69 IoU (XD_XD algorithm on las Vegas) respectively.

Fully convolutional networks are a generalization over CNNs and were introduced by (Long, Shelhamer, and Darrell 2015). Given that they don't have fully connected layers they can manage different input sizes and are faster. U-Net (Ronneberger, Fischer, and Brox 2015) is an improved version of the original FCN that has in the upsampling part a large number of feature channels to better propagate context information to higher resolution layers. In this work we used a modification of U-Net that was implemented by Juha Reunanen and Davis King on (King 2009).

In brief, we use a CNN to segment the parking blocks from an satellite view. Even though quite simple, this approach has been never been taken in previous works at the moment of this writting, maybe due to the lack of a dataset that could serve for training.

Recall that our solution proposal consist of using the segmentation of a public satellite image and combine it with the surveillance camera view to generate an extremely accurate segmentation on the surveillance camera, regardless of any temporal or obstacles interfering with camera's view. An outline of the developed method is given below.

Input: Trained CNN model, a surveillance camera test image and its corresponding satellite view.

Parameters: Camera projection parameters, i.e. intrinsic and extrinsic calibration matrices.

1. Take photo on a satellite.
2. Crop to parking area and scale.
3. CNN residual network segments the image.
4. Photo of the surveillance camera.
5. Calculate the homography between the two images.
6. Final result of the camera image segmented.

See Figure 2 for a workflow diagram of these steps.

Output: Segmentation results on the test image.

The first two steps, i.e. (a) and (b) are explained further on Section 3.1. Also the procedure for (d), in which we collected the surveillance camera images, is included there.

For (c), the fully convolutional CNN architecture was inspired in the U-Net architecture (Ronneberger, Fischer, and Brox 2015) and implemented by Juha Reunanen and Davis King on (King 2009). Firstly, we randomly crop 227x227 chips, emulating the same methodology of AlexNet in the ImageNet ILSVRC-2012 challenge (Krizhevsky, Sutskever, and Hinton 2012). The chips were also randomly horizontally flipped. Then, the input layer was submitted to 220 layers. We also used skip connections in each block as suggested by (He et al. 2016).

3. Dataset

Our method uses a training set to adjust the parameters of the CNN from the satellite perspective. Our dataset has roughly 7000 polygons. Normally a classical CNN would require much more samples to train e.g., AlexNet was trained using 1.2 million training images from ImageNet. By using the fully convolutional architecture implemented by King (based on Ronneberger) we are able to circumvent this requirement and train with only 4034 parking block polygons i.e., 300 images. Ronneberger explains the importance of data augmentation for learning shapes with elastic deformations in their architecture.

3.1. Image Collection Procedure

APKLOT consists of a total of 500 high-resolution images with more than 7000 annotated polygons. The images were collected first by downloading the coordinates from Open Street Maps. Then, we downloaded the patches containing individual parking lots from google maps using a single zoom level. The parking lots were filtered according to the following conditions:

- They show outdoor parking lots.
- At clear orthogonal daylight.
- With relatively low humidity.
- Demarcated using international standards or the ones corresponding to that country for public parkings.
- Meant to harbor vehicles no larger than full-sized cars (no more than 5,350 mm of length).
- Full parking lot view if possible.
- At least a block of 3 parking spots must be visible.

Then, on Google Maps, the proper zoom can be calibrated to a specific meters per pixel ratio using the following formula[4]:

$$metersPerPx = 156543.03392 * \frac{\cos(latLng.lat() * \pi/180)}{2^{zoom}} \quad (1)$$

Equation 1 shows the formula for calculating the meters per pixel ratio. $latLng.lat()$ is the object that contains the latitude as float for the specified location. In our case we set the zoom level to 20 for 0.1407m/pixel (roughly 14cm per pixel).

Cropping was made by taking the following considerations in mind:

- The resulting image must have the same zoom as the images in the training set, *regardless of its width and height.*
- The parking space preferably must be on the center of the image.
- Edge structures that do not correspond to parking spots and traffict lanes can be safely excluded.

[4]taken from (Stackexchange 2017)

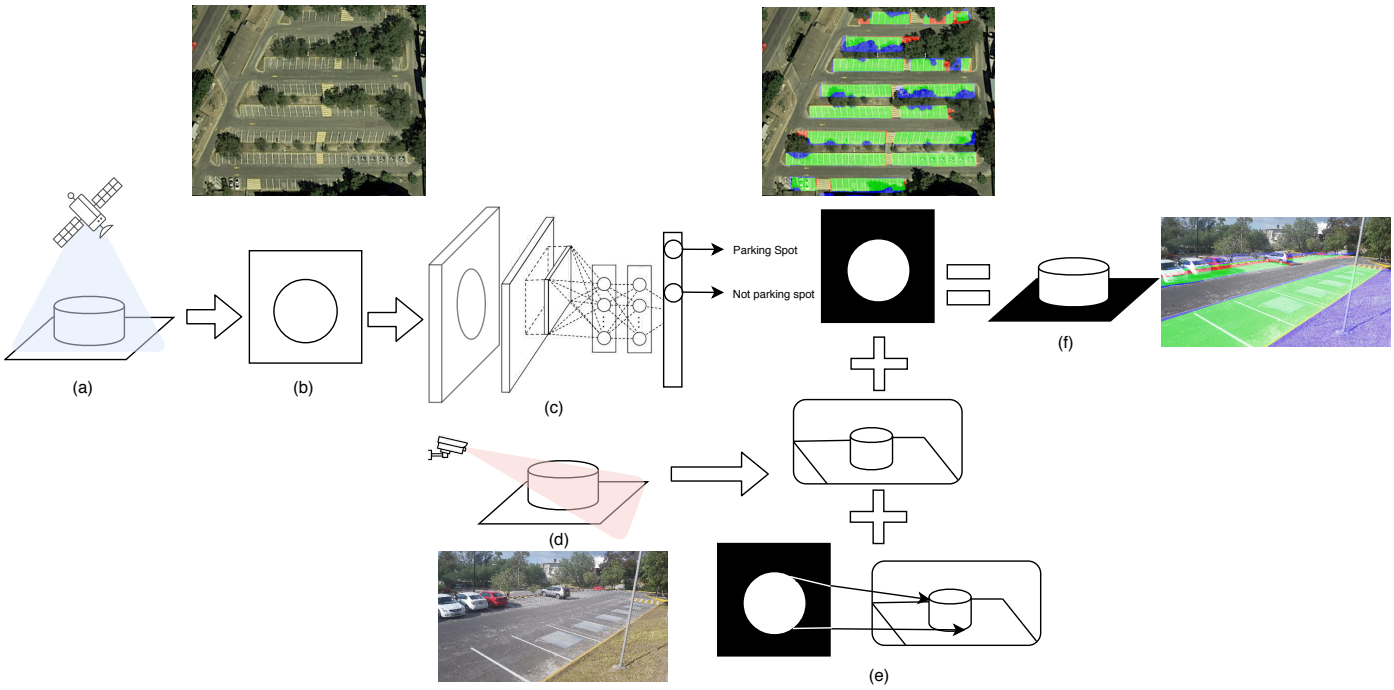


Figure 2: Our Proposal. (a) Photo of the satellite perspective, (b) Crop to parking area and scale, (c) CNN residual network segments the image, (d) Photo of the surveillance camera perspective, (e) Calculate the homography between the two images, (f) Final result of the camera image segmented. Red overlay stands for false negatives, blue for false positives and green for true positives.

3.2. Annotated Attributes

The images were annotated by marking the parking block polygons using the labelme tool. Complete parking blocks were annotated i.e., we are not considering the following:

- Parking spots outside the parking lot.
- Badly parked vehicles, including those parked on the traffic lane and non-parking spot (benches, gardens, etc).
- Debris or machinery in the parking spot when it is used as manner of storage facility.
- Trees in the way of the parking spot.

These annotations where then transformed into the Pascal VOC that consist in the following folder: JPEGImages, Annotations, ImageSets\Segmentation, SegmentationClass and SegmentationObject. We are also providing the original labelme format.

3.3. Dataset Statistics

Table 1 shows the control figures for each of the produced image sets.

Figure 3 shows general statistics of the input and mask coverage ratio. We can see the annotated class is clearly unbalanced, with only 24.5% corresponding to parking blocks. To overcome this challenge we must select a metric that compensates this phenomenon. In the next section we will see how the intersection over union (IoU) metric has been used efficiently in the past for similar cases.

Table 1: Image sets in *APKLOT*

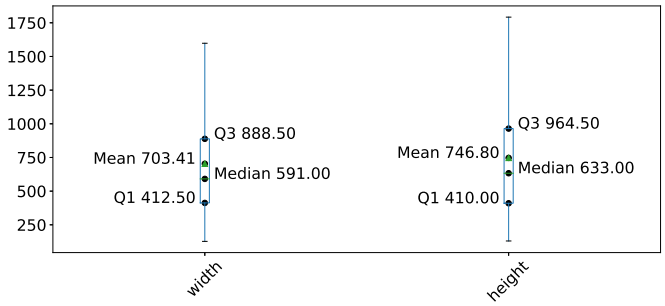
Year	Statistics	Notes
2018	Only 1 class discriminating between parking spot and other spaces. Train: 300 images 4034 labelme polygons Validation: 100 images 1513 labelme polygons Test: 101 images 1459 labelme polygons	Images were taken from Google Maps API at zoom level 20.

3.4. Evaluation Protocol

This section describes an evaluation protocol proposed on the basis of the APKLOT dataset. This protocol is the applied further on in the experiments section.

Overall accuracy metric considers the model's capability of not including some areas in the predicted class, i.e., the negative cell on the contingency table. Conversely, recall, precision and the F-Measure are considered to be biased in this sense. In order to avoid this bias, the authors of PKLot (Almeida et al. 2015) and DLib (King 2009) opted to use only *overall accuracy*. Overall accuracy is given by Equation 2.

(a) input dimensions in pixels



(b) mask coverage in pixels

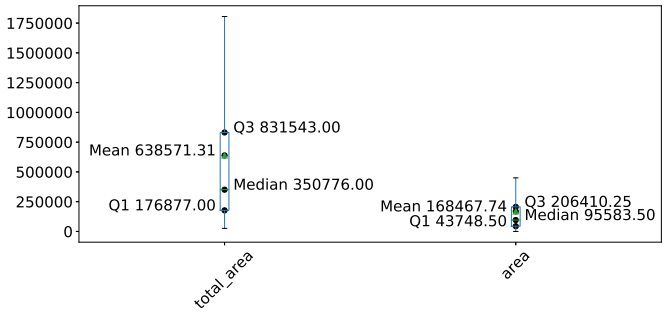


Figure 3: Size and sparsity statistics of the input images.

$$Acc = \frac{T_p + T_n}{P + N} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} = \frac{Right}{Right + Wrong} \quad (2)$$

This metric is usual in cases where we would like to classify individual instances without considering their spacial locality. Yet, the case of pixel segmentation deserves a special consideration. Each pixel is an instance we want to classify. However, most of the features come from the neighboring features. A foreground detected near the ground truth has presumably more informational value than a background detected near the ground truth to such a point that the distinction between foreground and background becomes relevant.

One of the Pascal VOC (Everingham et al. 2010) challenges consist on detecting many classes and marking them by using bounding boxes. Then, the background coincidences play no important role in how to position and scale the bounding boxes over detections. However, the overlapping between the true box and the predicted box is a good indicator of how well the prediction is performing.

$$IoU = \frac{T_p}{T_p + F_p + F_n} = \frac{\bigcap_{area}}{\bigcup_{area}} \quad (3)$$

Equation 3 shows the IoU formula proposed by (Everingham et al. 2010). It measures the overlapping pixels between the ground truth polygon and the prediction polygon, regardless of the similarity between the areas outside of the polygons. That is, the true negatives are not taken into account. The idea is to reward overlapping foregrounds, instead of backgrounds. An IoU score greater than 0.5 is normally considered a "good" prediction e.g., 10 different IoU thresholds are considered from 0.5 to 0.95 in the COCO challenge (Lin et al. 2014).

4. Experiments and Results

Having reviewed the structure of *APKLOT*, lets see how it is used on the steps required by the methodology.

Firstly, we decided to compare the training of the CNN with the original APKLOT dataset to training with an augmented version of APKLOT i.e., the 300 satellite images of the original training set were used to train one set of experiments. In parallel, 100 samples for each of those images were generated to accrue a total of 30300 sample images for a second set of experiments. In particular the data augmentation that was done consisted in the following actions:

- Randomly crop by a value between 0 and 50 pixels
- Horizontally flip 50% of the images
- Vertically flip 50% of the images
- Rotate images by a value between -45 and 45 degrees

Secondly, testing of the CNN was done with 100 independent satellite view images (hereby ___World_). In parallel, 14 independent satellite view images of the Tecnologico de Monterrey (ITESM) were also segmented. These images are the ones we will use to test the algorithm performance once we have translated the results into the 62 surveillance camera images inside ITESM.

Finally, the homographies per each of the 62 surveillance camera perspective images are calculated from the correspondence points and applied to the previous 14 images segmentation results. To find the best four points for the homography, we used algebraic error, default for the OpenCV `findHomography` method. See Figure 4 for a visual guide of these steps.

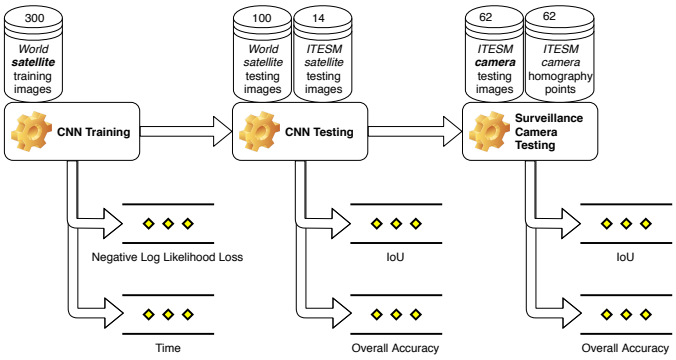


Figure 4: *APKLOT* dataflow between training, testing and translating the results to the surveillance camera image. Proposed metrics for measuring success are shown below each step. These metrics are discussed under the *Evalutation Protocol* section.

Figure 4 shows a tree structure to help us visualize the data flow and metrics used. The yellow diamonds correspond to the main metrics for measuring the contribution of

this work. We already explained both overall accuracy and intersection over union in Section 3.4. Multiclass logistic regression (i.e., negative log-likelihood loss, NLL or categorical cross entropy) is used in tandem with a softmax layer (one-hot) and estimates the maximum a posteriori class label given the parameters learned by the CNN. Each of the dataset icons is captioned with the number of annotated instances contributed.

4.1. Convolutional Neural Networks Setup and Training

In this section we describe the experimental setup, mainly for training the satellite segmentation CNN. Table 2 shows the hyperparameter values and significance for generating the experiments.

Table 2: Hyperparameter experimental setup.

Hyperparam.	Value	Ab.	Explanation
sample size	300 initial instances and 30300 instances by augmentation	j	More data is always the best way of reducing the variance without increasing the bias.
batch size	{16,30,32}	b	Using the full instance always gives better accuracy than using a randomly partitioned batch size. However, the model evaluates the full gradient on each epoch and it making it too expensive to train. Using minibatches can ameliorate this problem, although the exact size cannot be taken for granted to be the largest because we are dealing with an stochastic way of partitioning the data. We tuned up this parameter empirically.
initial learning rate	0.1-0.01	l	This value was set taking into account the recommendation of (Bengio 2012; Smith 2017)
minimum learning rate	0.00001-0.000001	r	A smaller learning rate produces more stable hops in the gradient on the seek of optimal weight parameters. Too small learning rate bogs down the convergence speed, though.

Ancillary techniques were used such as *momentum* with *weight decay* and *batch normalization* . The values for momentum were set according to (Sutskever et al. 2013) (0.9 momentum and 10^{-4} weight decay). Batch normalization window was set large enough to preserve a maximum convolutional layer of 512.

Finally, the experiments were named using the following naming convention:

*<jittering was used>***j-***<batch size>***b-***<initial learning rate>***l-***<minimum learning rate>***r**

In total we made 15 experiments. 12 of these used external data augmentation already described in the previous section besides the one that does dlib by default (random cropping and horizontal flipping). The other experiments only used the dlib default data augmentation. In the data augmentation experiments initial learning rate was set fixed to 0.1 in contrast to the cases not using data augmentation that also used 0.01. However, batch sizes were cycled equally between the three values 16, 30 and 32.

At the first experiments we found that minimum learning rate was unimportant so the experiments corresponding to a batch size of 32 use only 10^{-5} in contrast to the 16 and 30 cases that use also a 10^{-6} minimum learning rate.

During training we found that models using data augmentation produced a higher negative log likelihood with a p-value of 0.6% (Mann-Whitney-U). Also we found that models with a batch size less than 16 had a shorter duration with a p-value of 0.4%. The faster model was using rotation jittering, 16 batch size, 0.01 initial learning rate and minimum learning rate of 0.000001. See Figure 5 for a boxplot of these results, *(a)* shows the negative log likelihood and *(b)* the durations.

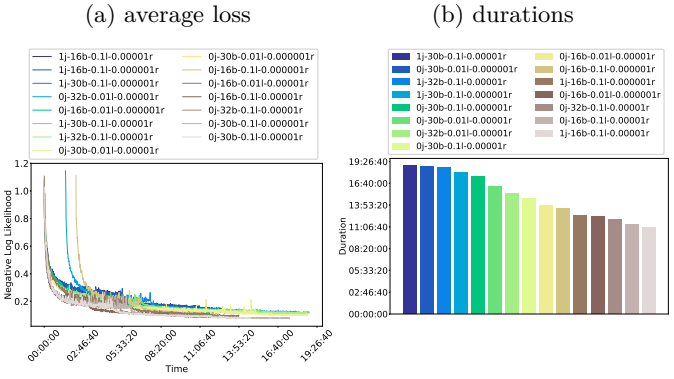


Figure 5: The main features for measuring progress are shown: *(a)* lesser average loss increases the possibility of reaching the 5000 steps without progress limit, *(b)* overall durations, the top is about 19 and a half hours.

Each of these experiments were tested using a I7 Intel Core with 32GB of RAM and 11GB GPU, namely using a NVIDIA GeForce GTX 1080 Ti. Images more than 4,000,000 squared pixels were too big for the GPU and triggered a CUDAMalloc (memory allocation) error.

4.2. Expected Results

Each of the perspectives we are using is subject to error sources that could damage our results. On the satellite segmentation step we can have occlusion, objects with very similar texture to parking blocks and no visible parking lines at all. See Figure 6 for a visual reference of these errors.

Cases *(a)* and *(b)* could be solved using a Bayesian approach (like the one used by Huang and Wang) to learn the overall structure of the parking lot and discarding the areas that deviate to much from the learned parameters. Case *(c)* is a little bit trickier, to solve this case we would need to change our selection

(a) (b) (c)

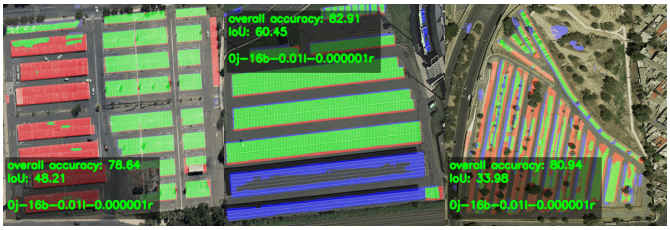


Figure 6: Parking lot examples from satellite perspective in which our method is prone to fail: *(a)* transparent roof, *(b)* grid like roof, *(c)* no visible demarcation lines. Red overlay stands for false negatives, blue for false positives and green for true positives.

criteria in the dataset and include more images of unmarked but *somehow* identifiable parking lots.

On the camera perspective we can have resolution mismatch and multilevel parking lots. Subfigures *(a)* and *(b)* of Figure 7 show these errors respectively.

(a) (b)

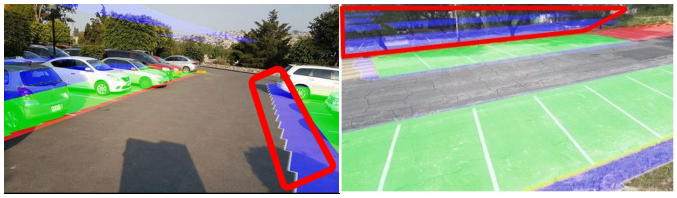


Figure 7: Parking lot examples from a camera perspective in which our method is prone to fail: *(a)* resolution mismatch, *(b)* multilevel parking lot. Red overlay stands for false negatives, blue for false positives and green for true positives. A red outline is shown around the aforementioned problems.

Case *(a)* could be solved by trivially calculating a linear interpolation of a convex polygon shape. Case *(b)* could be solved by integrating height data (Wang and Hanson 1998) to separate planes in the homography calculation.

4.3. Image Segmentation Results

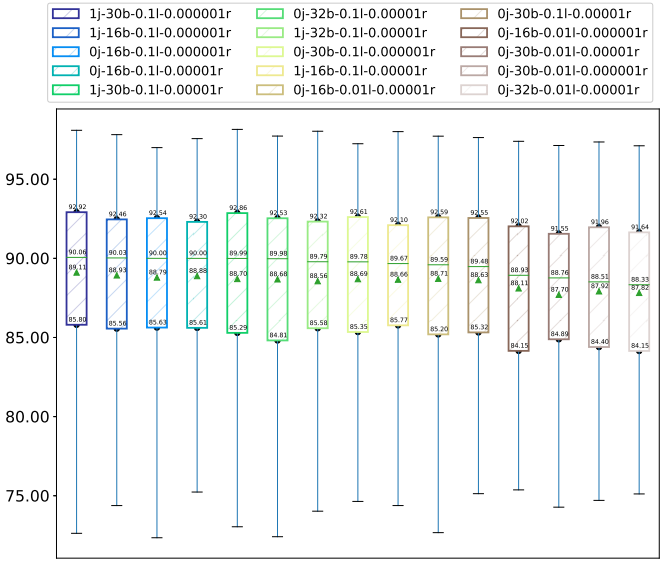
In this section we present the results from testing both segmentation on the satellite 100 independent images and on the surveillance camera images located at the ITESM. We are presenting both results to allow improvements along the pipeline.

Firstly, in the satellite image segmentation we achieved more than 50% intersection over union in all models, shown in Figure 8. Subfigure *(a)* show the results using overall accuracy. This metric achieved a notably higher score than the intersection over union shown in Subfigure *(b)*. Recall that in intersection over union we are not considering true negatives, just true positives.

The variance of the models can be seen from the standard deviation. The most stable model (less standard deviation) was *0j-16b-0.01l-0.000001r* which achieved a median of 88.93 overall accuracy. The model scored second if we order them by the worse segmented example, so this model can be considered the best among those we trained.

Satellite segmentation results on the ITESM images (see Figure 9) had greater variance. Nonetheless, the median remained

(a) overall accuracy



(b) intersection over union

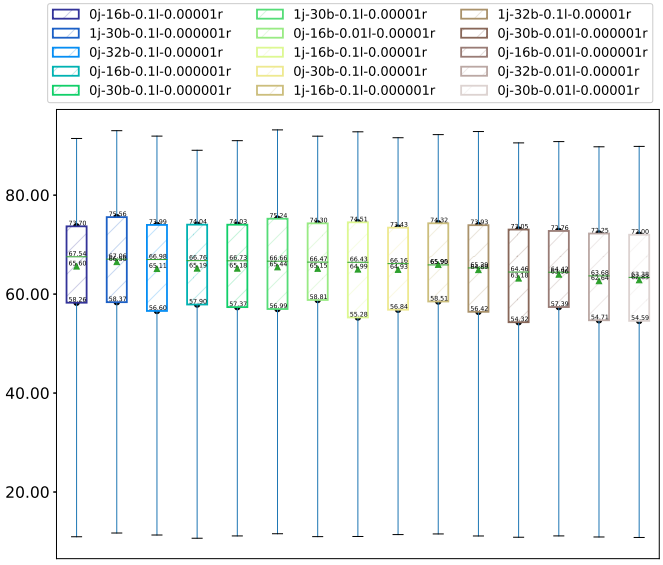
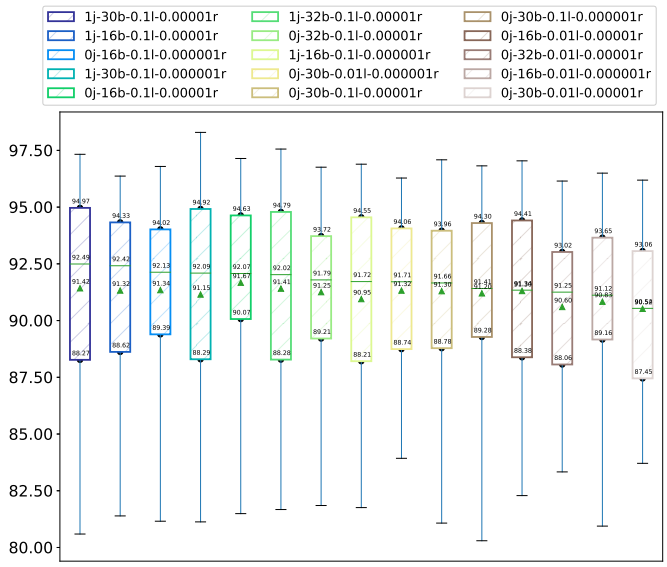


Figure 8: Segmentation satellite results on the *World* dataset (100 images): *(a)* is by using dlib's measure - overall accuracy, *(b)* shows intersection over union. The bests model considering the medians (the mean is vulnerable to outliers) are for *(a)* 1j-30b-0.1l-0.000001r and for *(b)* 0j-16b-0.1l-0.00001r.

inside two orders of magnitude of the previous results. Wilcoxon signed rank test showed that we can reject the null hypothesis that the two medians are the same. There is a significant improvement in the ITESM images using IoU. This could be explained by the absense of roof like structures in ITESM that can produce a greater number of false positives. Also, parking lots are very well maintained and painted inside the institution in contrast with the *World* dataset. Counterintuitively, the results are inverted when using overall accuracy and there is yet a significant difference. We surmise that there are much more true negative areas, meaning empty spaces on the images of the *World* dataset. Also, given the ITESM location on a top-hill

(a) overall accuracy



(b) intersection over union

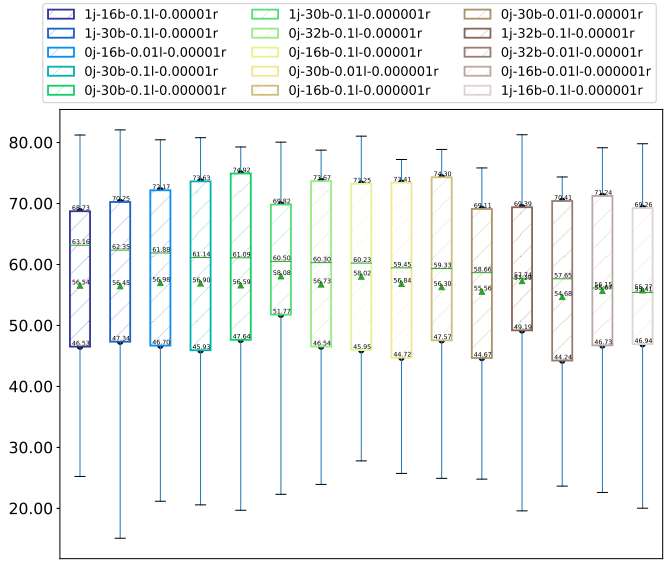
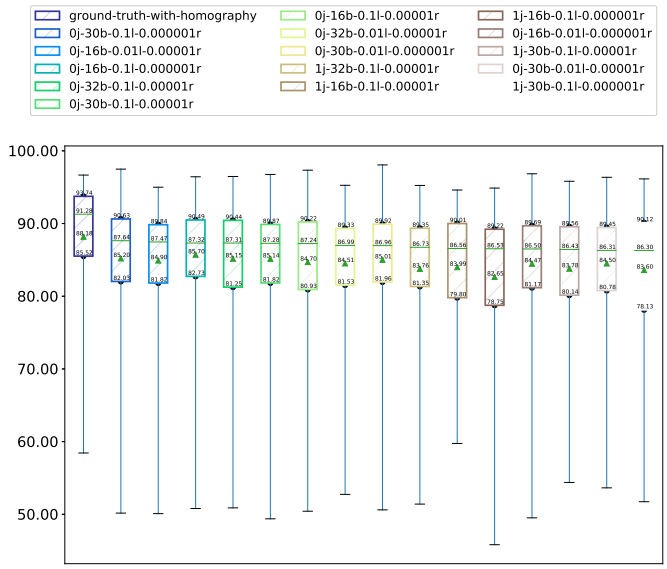


Figure 9: Segmentation satellite results on the *ITESM* dataset (14 images): *(a)* is by using dlib's measure - overall accuracy, *(b)* shows intersection over union. The bests model considering the medians (the mean is vulnerable to outliers) are for *(a)* 1j-30b-0.1l-0.000001r and for *(b)* 0j-16b-0.1l-0.00001r.

(a) overall accuracy on surveillance camera perspective



(b) intersection over union on surveillance camera perspective

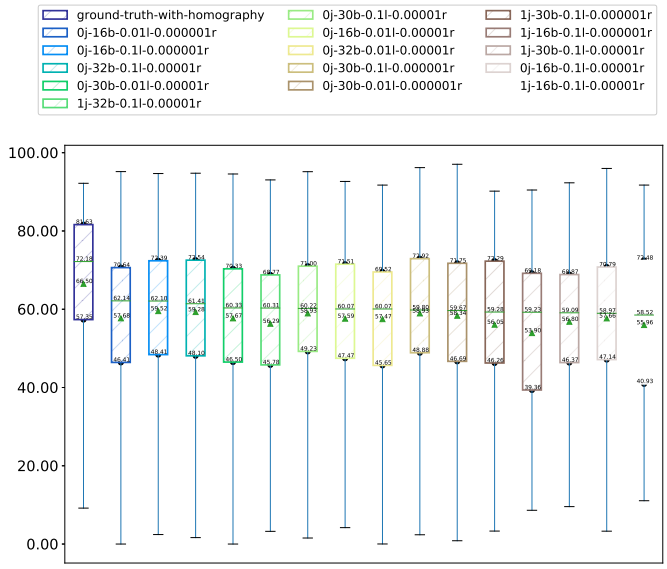


Figure 10: Segmentation **camera** results on the *itesm* dataset (14+62 images): *(a)* is by using dlib's measure - overall accuracy, *(b)* shows intersection over union. The bests model considering the medians (the mean is vulnerable to outliers) are for *(a)* 0j-30b-0.1l-0.000001r and for *(b)* 0j-16b-0.01l-0.000001r. Satellite ground truth homography transformation to the camera perspective is provided for reference.

there is also a lot of variability in the slopes that increase the false positive rate.

Segmentation on the satellite images preformed quite well both in time and accuracy. However, when introducing an homography transformation the results dropped, shown in Figure 10. Considering this fact we can safely conclude that any improvement in the homography calculation will greatly improve the overall result.

All segmentation done in *APKLOT* follows a clear rule: mark the area wherever is visible. However, one of the known issues of evaluating a supervised segmentation task is that the ground truth data is vulnerable to the subjective criteria of each person (Zhang, Fritts, and Goldman 2008). Furthermore, pixel-by pixel segmentation have many challenges like: partial occlusions, tool limitations, resolution mismatches and alike. Considering these natural limitations, a perfect accuracy would only be achievable in case the two shapes i.e., the predicted and the ground truth shapes, were identical. To overcome this challenge, we have included the first boxplot of Figure 10 as baseline. This boxplot shows the results of the satellite ground truth images with the applied homography. Finally, all the boxplots were evaluated using the real ground truth segmented by a human in the camera perspective.

5. Conclusions and Recommendations

The main focus of this paper was solving the parking block segmentation on a surveillance camera perspective problem by taking advantage of the satellite view of the same scene. The goal of this research was two-fold. Firstly, to segment parking blocks on aerial images. Given the inexistence of a proper dataset to accomplish this specific task, we proposed *APKLOT*, a collection of 7000 parking block polygons that proved enough to achieve 50% IoU world wide. Secondly, to translate these results to a surveillance camera perspective and measure the ensuing degradation. By testing on several CNNs we provide groundwork to support the relationship between each hyperparameter, learning rate and duration. We demonstrated that using a simple architecture like the U-Net with skip connections is sufficient and enough to segment simple shapes like the ones composing parking blocks even though many are heavily occluded.

For future work we recommend to learn the homography automatically without relying on explicitly providing the camera projection parameters or the correspondence points. Then, our approach could be applied to any outdoor parking lot without human intervention. For the case of automating the correspondence points extraction, they can be learned using a metric learning approach as proposed in (Altwaijry et al. 2016).

6. Acknowledgement

We wish to thank the *Instituto Tecnologico y de Estudios Superiores de Monterrey* for financing the Open Access publication and providing the surveillance camera images. N. Hurst gratefully acknowledges the scholarship from CONACyT to pursue his postgraduate studies and making this research possible.

References

Almeida, Paulo R.L., Luiz S. Oliveira, Alceu S. Britto, Eunelson J. Silva, and Alessandro L. Koerich. 2015. "PKLot - a Robust Dataset for Parking Lot Classi Cation the Pklot Dataset." *Expert Systems with Applications* 42 (11). Pergamon Press, Inc.: 1–6.

Altwaijry, Hani, Eduard Trulls, James Hays, Pascal Fua, and Serge Belongie. 2016. "Learning to Match Aerial Images with Deep Attentive Architectures." In *The Ieee Conference on Computer Vision and Pattern Recognition (Cvpr)*.

Amato, Giuseppe, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Carlo Meghini, and Claudio Vairo. 2017. "Deep Learning for Decentralized Parking Lot Occupancy Detection." *Expert Systems with Applications* 72: 327–34. https://doi.org/https://doi.org/10.1016/j.eswa.2016.10.055.

Bengio, Yoshua. 2012. "Practical Recommendations for Gradient-Based Training of Deep Architectures." In *Neural Networks: Tricks of the Trade*, 437–78. Springer.

Cheng, Liang, Lihua Tong, Manchun Li, and Yongxue Liu. 2014. "Extracting Parking Lot Structures from Aerial Photographs." *Photogrammetric Engineering & Remote Sensing* 80 (2). American Society for Photogrammetry; Remote Sensing: 151–60.

Cisek, D., M. Mahajan, J. Dale, S. Pepper, Y. Lin, and S. Yoo. 2017. "A Transfer Learning Approach to Parking Lot Classification in Aerial Imagery." In *2017 New York Scientific Data Summit (Nysds)*, 1–5.

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. "The Cityscapes Dataset for Semantic Urban Scene Understanding." In *The Ieee Conference on Computer Vision and Pattern Recognition (Cvpr)*.

Demir, Ilke, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raska. 2018. "Deepglobe 2018: A Challenge to Parse the Earth Through Satellite Images." In *2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition Workshops (Cvprw)*, 172–17209. IEEE.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. "The Pascal Visual Object Classes (Voc) Challenge." *International Journal of Computer Vision* 88 (2): 303–38.

Harrell, Erika. 2012. "Violent Victimization Committed by Strangers, 1993-2010." *Bureau of Justice Statistics Special Reports*, December. Bureau of Justice Statistics.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Identity Mappings in Deep Residual Networks." In *Computer Vision – Eccv 2016*, edited by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, 630–45. Cham: Springer International Publishing.

Hsieh, Meng-Ru, Yen-Liang Lin, and Winston H. Hsu. 2017. "Drone-Based Object Counting by Spatially Regularized Regional Proposal Network." In *The Ieee International Conference on Computer Vision (Iccv)*.

Huang, Ching-Chun, Sheng-Jyh Wang, Yao-Jen Change, and Tsuhan Chen. 2008. "A Bayesian Hierarchical Detection Framework for Parking Space Detection." In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE.

Huang, C., and S. Wang. 2010. "A Hierarchical Bayesian Generation Framework for Vacant Parking Space Detection." *IEEE Transactions on Circuits and Systems for Video Technology* 20 (12): 1770–85.

Huang, T., D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber. 1994. "Automatic Symbolic Traffic Scene Analysis Using Belief Networks." In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, 966–72. AAAI'94. Menlo Park, CA, USA: American Association for Artificial Intelligence.

Jung, Ho Gi, Dong Suk Kim, Pal Joo Yoon, and Jaihie Kim. 2006. "Parking Slot Markings Recognition for Automatic Parking Assist System." In *2006 Ieee Intelligent Vehicles Symposium*, 106–13.

Kabak, Mehmet Ozan, and Ozhan Turgut. 2010. "Parking Spot Detection from Aerial Images." *Stanford University, Final Project Autumn 2010, Machine Learning Class*.

King, Davis E. 2009. "Dlib-Ml: A Machine Learning Toolkit." *Journal of Machine Learning Research* 10: 1755–8.

Koutaki, Gou, Takamochi Minamoto, and Keiichi Uchimura. 2016. "Extraction of Parking Lot Structure from Aerial Image in Urban Areas." *International Journal of Innovative Computing Information and Control* 12 (2). ICIC International Tokai University, 9-1-1, Toroku, Kumamoto, 862-8652, Japan: 371–83.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing*

Systems 25, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, 1097–1105. Curran Associates, Inc. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. "Microsoft Coco: Common Objects in Context." In *European Conference on Computer Vision*, 740–55. Springer.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. "Fully Convolutional Networks for Semantic Segmentation." In *The Ieee Conference on Computer Vision and Pattern Recognition (Cvpr)*.

Marmanis, D., K. Schindler, J.D. Wegner, S. Galliani, M. Datcu, and U. Stilla. 2018. "Classification with an Edge: Improving Semantic Image Segmentation with Boundary Detection." *ISPRS Journal of Photogrammetry and Remote Sensing* 135: 158–72. https://doi.org/https://doi.org/10.1016/j.isprsjprs.2017.11.009.

Marmanis, D., J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla. 2016. "SEMANTIC Segmentation of Aerial Images with an Ensemble of Cnns." *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* III-3: 473–80. https://doi.org/10.5194/isprs-annals-III-3-473-2016.

Mexas, Antonio H, and Maurício Marengoni. 2014. "Unsupervised Recognition of Parking Lot Areas," no. 68 (February). Avestia.

Murphy, Kevin P. 2013. *Machine Learning : A Probabilistic Perspective*. Cambridge, Mass. [u.a.]: MIT Press.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–41. Springer.

Sastre, R. J. L., P. G. Jimenez, F. J. Acevedo, and S. M. Bascon. 2007. "Computer Algebra Algorithms Applied to Computer Vision in a Parking Management System." In *2007 Ieee International Symposium on Industrial Electronics*, 1675–80.

Seo, Young-Woo, Nathan D Ratliff, and Chris Urmson. 2009. "Self-Supervised Aerial Image Analysis for Extracting Parking Lot Structure." In *International Joint Conferences on Artificial Intelligence*, 1837–42.

Sevillano, X., E. Mármol, and V. Fernandez-Arguedas. 2014. "Towards Smart Traffic Management Systems: Vacant on-Street Parking Spot Detection Based on Video Analytics." In *17th International Conference on Information Fusion (Fusion)*, 1–8.

Smith, Leslie N. 2017. "Cyclical Learning Rates for Training Neural Networks." In *Applications of Computer Vision (Wacv), 2017 Ieee Winter Conference on Applications of Computer Vision*, 464–72. IEEE.

Stackexchange. 2017. "What Ratio Scales Do Google Maps Zoom Levels Correspond to?" Geographic Information Systems Stack Exchange. 2017. https://gis.stackexchange.com/questions/7430/what-ratio-scales-do-google-maps-zoom-levels-correspond-to/7443.

Sutskever, Ilya, James Martens, George Dahl, and Geoffrey Hinton. 2013. "On the Importance of Initialization and Momentum in Deep Learning." In *Proceedings of the 30th International Conference on Machine Learning*, edited by Sanjoy Dasgupta and David McAllester, 28:1139–47. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR. http://proceedings.mlr.press/v28/sutskever13.html.

Wang, Xiaoguang, and Allen R Hanson. 1998. "Parking Lot Analysis and Visualization from Aerial Images." In *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV98 (Cat. No.98EX201)*, 36–41.

Weis, Tim, Benjamin May, and Christian Schmidt. 2006. "A Method for Camera Vision Based Parking Spot Detection." In *SAE Technical Paper Series*. SAE International.

Zhang, Hui, Jason E. Fritts, and Sally A. Goldman. 2008. "Image Segmentation Evaluation: A Survey of Unsupervised Methods." *Computer Vision and Image Understanding* 110 (2): 260–80. https://doi.org/https://doi.org/10.1016/j.cviu.2007.08.003.