Semantic Segmentation of RGB Aerial Images Using CNN and Superregions

Mariana-Edith Miranda-Varela, L. Oyuki Rojas-Perez, and Jose Martinez-Carranza*

Instituto Nacional de Astrofisica, Optica y Electronica (INAOE), Puebla, Mexico

ABSTRACT

In recent years, large amounts of aerial images are readily available because they are taken from aeroplanes, hot-air balloons, and unmanned aerial vehicles. However, these images are not relevant if their information is not interpreted. Although there are several techniques to analyse them, i.e., from statistical methods to machine learning techniques, at present the most popular technique is deep learning, where the convolutional neural network (CNN) has shown good accuracy and success on several studies of satellite images with a ground sampling distance $>$ 10 m. In this work, we employ red-green-blue (RGB) aerial images taken from 100-300 meters above the ground, which are employed to build a CNN based on a variant of a VGG16 architecture, for the semantic image segmentation. Four classes are identified, urban zone, vegetation zone, agricultural zone and roads. Due to the high spatial resolution, pixels are grouped with respect to similar texture in superregions to reduce the number of pixels classified by the CNN. Our approach is able to segment an aerial image in three modalities: i) pixel-wise segmentation; ii) superpixels, this is, a group of pixels; iii) superregions, which is a group of superpixels. The model is tested with aerial images with a height between 200 to 400 meters. According to our results, the accuracy is similar for each one of the three methods presented in this work. However, the time performance is significantly reduced when employing superpixels and superregions. Furthermore, we have obtained an average F-score 0.732, comparable to state-of-the-art approaches.

1 INTRODUCTION

At present, the interpretation of satellite and aerial image is an essential task, because this kind of images provides important data to bring about land cover map, land use map, and object detection. The land cover and land use maps can be used as a tool to plan a city [1] or to monitor either deforest or changes of a particular region [2]. Moreover, this information can be recorded to form a geographical database. Concerning object detection, some types of targets can be vehicles, buildings, roads [3], aeroplanes [4], and so on.

Semantic segmentation is part of visual computation that assigns a classification label to each pixel of an image. There are many algorithms to segment an image, which are classified in: threshold-based, edge-based, region-based and classification-based methods, where the last is able to get more accuracy values with appropriate feature extractors and classifiers [5]. Deep learning models are used to classify or segment an image, where Convolutional Neural Network (CNN) reported the best performance. A CNN is a single model, for this reason it is regarded as a one-step method [5].

To build a CNN model for semantic segmentation is necessary a dataset, which is employed to train and test the model. Some dataset available online are ISPRS benchmark[1], Google Maps, Linz Data Service[2], among others. Generally, these are composed of high-resolution orthophotographs with a ground sampling distance (GSD) of $>$ 10 cm [6]. Most of the studies of semantic segmentation using CNN employ mainly the ISPRS benchmark, that makes up of high-resolution images and the digital surface map (DSM), where the images consist of near infrared (NIR), red and green bands, and they have a GSD of less than 10 cm. The NIR can be used to compute the Normalized Difference Vegetation Index (NDVI) and the Green Normalized Vegetation Index (gNDVI) using the red and the green band, respectively. Both indexes are utilized to distinguish easily the vegetation into an image [7]. Furthermore, DSM provides height information, that improves the precision of the semantic segmentation [8–10].

At present, most of the research on semantic segmentation has focused on producing a CNN model through high-resolution orthophotographs and its DMS. In contrast to, the CNN models build by means RGB images have been less explored. Moreover, in this work, we consider aerial images taken from a lower distance than the high-resolution orthophotographs, that is between 100 and 300 meters above the ground. Thereby these images contain small area in greater detail. As far as we know, there is not approach to build

*Email address(es): carranza@inaoep.mx

[1] http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html

[2] https://data.linz.govt.nz/

a CNN model using RGB aerial images taken close to the ground. The classification classes for our work are roads, agricultural zone, urban zone and vegetation zone.

In this paper, a CNN model is produced through raw RGB aerial images to achieve a semantic segmentation at pixel-wise classification. To take advantage of the high spatial resolution of images, preprocessing techniques to group pixels similar in colour are employed, and, thereby the number of pixels classified is reduced. The first technique yields superpixels, which are grouped based on their average colour to constitute superregions. The structure of the paper is as follows: Section 2 describes the related work. Section 3 details the implemented system. In Section 4 the experimental results are presented and in Section 5 the conclusions and future work are exposed.

2 Related Work

Semantic segmentation is a challenging task for satellite and aerial images, because of the variety of information on an image, such as texture or illuminance. On the one hand, different types of textures for buildings or vegetation are present. On the other hand, the relative absence of light, shadows, are a source of noise to classify an aerial image correctly. There are many techniques to segment this kind of images, although those works that employed a deep learning model have reported the best performance [11].

SegNet has proved to get better results for segmenting images [12]. Thereby, this architecture has been employed in several works for semantic segmentation of satellite and aerial images. [13] extended SegNet architecture to distinguish eight classes: building, sealed area, bare soil, grass, tree, water, car and other on multi-spectral images (RGB + NIR). Besides, DSM and Digital Terrain Model (DTM) were used to represent the height information of the experiment region. In [11], a SegNet variant was proposed to identify roads, buildings, low vegetation, trees and cars on NIRRG images. Apart from the dataset, DSM, Normalized DSM (NDSM), and NDVI were used to improve the final prediction.

One of the most important benchmarks for semantic segmentation of aerial images is ISPRS, which has been employed in the following approaches. [14] proposed a Local Attention Network (LANet), composed by a patch attention module (PAM) and an attention embedding module (AEM), meanwhile, PAM improves the local context information, AEM enhances the use of spatial information. [15] solved it by means a multi-skip convolutional network and Markov Random Fields. The former extracts the context information and the latter refines the results. Although their segmentation is effective, the time required to get it is high. In the aforementioned approaches, the categories identified were: roads (impervious surfaces), building, low vegetation, tree, car and clutter/background. [10] solved the semantic segmentation utilizing CNN and hand-crafted features, in the end, a Conditional Random Field (CRF) was applied. They identified five classes: impervious surfaces, building, low vegetation, tree, and car. Moreover, the pretrained model was tested on another type of data. The best performance was got with NIRRG images, DSM and NDSM. In [16] a Fully Convolutional Networks (FCNs) is proposed, which has two identical CNN that are merged before the final layer. The input for each CNN is the image and the digital elevation model (DEM), respectively. TreeSegNet is proposed in [17], which is composed on three elements: (1) the segmentation module, that is, an encoder-decoder architecture; (2) the Tree-CNN block, which improves the result of the previous element and it is the centre part of this approach, and (3) the concatenating connections. This approach was trained and tested through five channels R, G, B, NIR and DMS. [18] proposed an encoder-decoder architecture with skip connections, and Fully connected CRF (FCRF) post-processing, which improved the results. The input CNN model was the NIRRG images and NDSM from ISPRS benchmark. In [19], a hybrid FCN architecture is proposed, which merge the image channels with DSM. Another approach based on CNN is presented in [9], where CNN was tested on multispectral orthoimageries and DSM to classify vegetation, ground, roads, buildings and water.

According to the results of previous works, semantic segmentation using CNN reports a high level of accuracy, moreover, the use of the DMS improved the performance measures [8–10]. However, DSM is more common for satellite images than aerial images and the NDVI can be computed only in those images with NIR channel. Due to the semantic segmentation utilizing only RGB aerial images has been disregarded, in this work builds a CNN model through RGB aerial images without extra information, such as DMS or NDVI. Moreover, the images of interest are those taken from 100 and 300 meters above the ground.

3 Methodology

In this section, the elements employed in our approach for segmenting aerial images are presented. Our proposal identifies four classes: agricultural, urban, vegetation and roads. Firstly, the dataset employed to train the model is exposed. Then, CNN architecture is described in detail. Finally, the types of classification are explained.

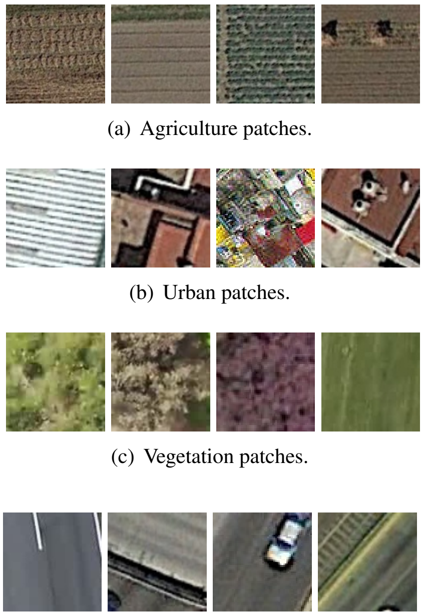
3.1 Dataset

The set of images employed to train the model is composed of 36000 images of size 60×60, named patches, which were extracted from Google Earth images correspond to the Megalopolitan area at the centre of Mexico[3] and pictures of different states of Mexico taken from drones. The altitude of the Google Earth images is between 200 and 300 meters, and aerial images taken by a drone from a range of height of 50

[3] Mexico City, the State of Mexico, Tlaxcala, Hidalgo, Morelos, and Puebla constitute the Megalopolitan area.

and 100 meters.

All aerial images were cropped into non-overlapping tiles of size 60×60 pixels, where each one has information corresponding to one class. The total of patches for each class is 9000. In Figure 1, four patches for each class are presented. Agricultural patches present information of sown, harvested and plantations grow land. Concerning urban patches, roof, roofing sheets, and objects that do not belong to the other classes. Information presented in vegetation patches is grass and trees. Finally, road patches contain roads and earthen roads.



(a) Agriculture patches.

(b) Urban patches.

(c) Vegetation patches.

(d) Road patches.

Figure 1: Example of patches for each class of the dataset.

Due to CNN requires a large amount of training data, the dataset was increased by the synthetic data augmentation technique. The applied transformation to the dataset was rotation and translation, with values of 45° and 90, respectively.

3.2 CNN architecture

There are several architectures for deep learning in the literature, however, the CNN have been applied in classification and semantic segmentation approaches with successful results [15]. CNN was inspired by the visual cortex organization of the living creatures [20], and it is composed by one or several CNN building blocks, that is, convolutional layer, activation function and pooling layer, followed by one or more fully connected layers and the loss layer [13].

The features of the input image are extracted by CNN layer through filters, named kernels, which are generally of size 3×3. This layer is the most important of the CNN building, its output is transformed employing an activation function. Following, several features are combined with the pooling layer. In this work the size of kernel used is 3×3 with stride 1. Moreover, the rectified linear unit (ReLU) is employed as the activation function, which preserves only the positive values, and the pooling method employed was max-pooling with a size of 2×2. In Figure 2 the CNN architecture employed is shown, which is based on VGG16 [21]. After the last block building CNN is employed one fully-connected layer, followed by one dropout layer with proportion 0.5, this is employed to prevent the overfitting and, finally, a softmax classifier which computes the probability for each class is added. The size of output vector is four which corresponds to the next classes: urban zone, vegetation zone, agricultural zone and roads. The input to the CNN model is a RGB patch of size 60×60, and the output is the label corresponding to the central pixel which is classified according to the rest pixels around it, this is considered as a patch-based approach. Our CNN model was trained from scratch with random initialization, with 100 epochs, and 80% of dataset were chosen randomly for training and the rest was employed for testing the CNN model. Furthermore, the base learning rate is set to 1E-5 with a batch size of 100.

Our approach can segment an aerial image of any size, however, the minimum size is 60×60. Due to the CNN model predicts the label of central pixel, this proposal is named pixel-based classification. As a consequence, the number of classified pixels is based on the size of the image, multiplying weight × height, which is a disadvantage for large images.

3.3 Superpixel-based classification

The pixel-based classification can produce a salt-and-pepper effect on the output image. This problem was tackled applying a simple segmentation technique, which groups adjacent pixels based on their colour and texture similarity [22], each group is named superpixel. This reduces the complexity of an image owing to the number of superpixels is lower than pixels. Therefore, the time required to segment an image is lower than pixel-based classification approach. Moreover, the appearance of spots is reduced. To extract superpixels of aerial image SLIC algorithm [23] is employed because this algorithm is faster than others superpixels algorithms and it has a good performance. SLIC is based on k-means using color and spatial information [22]. This approach is denoted as superpixel-based classification.

3.4 Superregion-based classification

Due to the lower distance considered for RGB images, can exist several superpixels with similar texture, which can be joined on large areas, denoted as superregion [24], also named super-segments. Therefore, a superregion is a connected set of adjacent superpixels, which have a bit difference of colour, that is computed by Eq. 1.

$$D_c = \sqrt{(r_j - r_i)^2 + (g_j - g_i)^2 + (b_j - b_i)^2} \qquad (1)$$

where D_c represents the Euclidean distance between i-superpixel and j-superpixel, which must be adjacent. r, g,

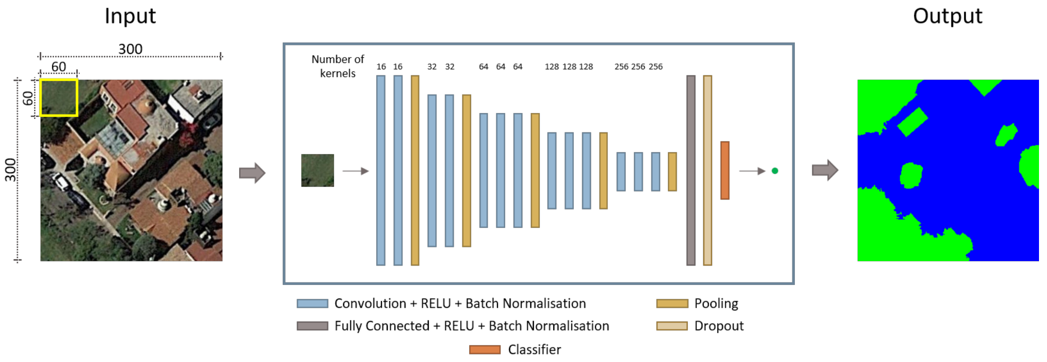


Figure 2: CNN architecture, which is composed by thirteen convolutional layers, one fully connected layer, a dropout layer and a classifier. On the top of each convolutional layer the number of kernels is shown.

and b are average of red, green, and blue values for i and j superpixels. To merge i and j superpixels, the D_c value must be less than a threshold.

An advantage of superregion is the time required to segment an image, which is lower than previous approaches because only a subset of superpixels for each superregion is classified, thereby, the number of superpixels classified is lower than the superpixel-based classification. Furthermore, the salt-and-pepper effect is reduced and the edges for each class are improved.

In Figure 3 an example of the original image for Image 1 (see Table 1), its version dividing into superpixel, where each superpixel is surrounded by a green line, and image dividing into superregion, whither each superregion is represented by different colour are shown. The number of pixels (Figure 3(a)) is 360000, meanwhile the number of superpixels (Figure 3(b)) is 1069 and the number of superregions (Figure 3(c)) is 421.

4 Experiments

In this section, the use of our model to segment aerial images is studied. The model was developed with libraries Keras and TensorFlow, and the experiments were conducted on a 2.6 GHz Intel Core i7 processor, RAM of 15GB and Nvidia GeForce GTX 970M GPU. In the following experiments, five aerial images are segmented, which are taken between 400 and 200 meters. The images, their size and their ground truth are presented in Table 1. Regarding ground true, roads are represented by brown colour, the vegetation zone is coloured with green colour, the urban zone is shown by blue colour and the agricultural zone is painted with yellow colour. With respect to Image 3, the architecture of the houses is different with respect to those employed to train the model.

The test images contain city and suburban surfaces, concerning the urban images present mixture textures for roofs, such as concrete, steel, and wooden. Meanwhile, the suburban environments are constituted by agricultural fields and vegetation regions, mainly.

4.1 Performance measures

In order to compute the accuracy of our model, three metrics are employed:

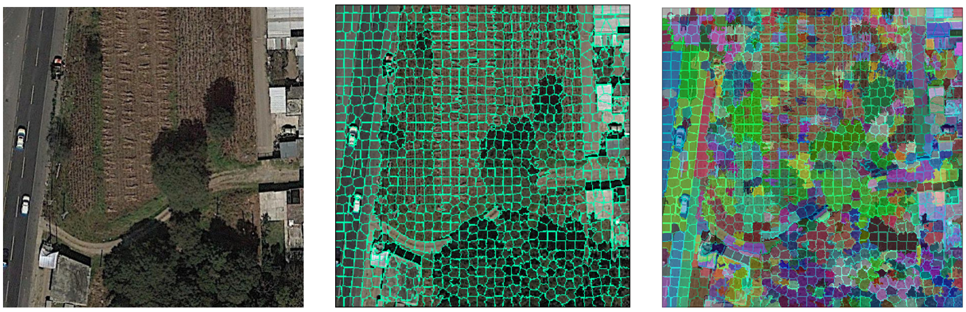
$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F\text{-}score &= \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}} \end{aligned}$$

where TP (True Positive) is the number of pixels correctly classified, FP (False Positive) is the number of pixels that according to CNN belong to one class but they really belong to another, and FN (False Negative) is the number of pixels that according to CNN do not belong to a class but they actually do. For each measure, its value is a real number between 0 and 1 and a value close to 1 is preferred. Consequently, $Precision$ is the ratio of the correctly classified pixels to all classified positive pixels, $Recall$ is the ratio of the correctly classified pixels to all actual positive pixels, and F-$score$, also called $F1$-$score$ or F measure, represents the balance between Precision and Recall.

The aim of this paper is to segment RGB aerial images employing a CNN model. For this, we adopted three schemes: pixel-based classification, superpixel-based classification and superregion-based classification, where the last two experiments considered a set of pixels with colour similarity.

For each experiment, the values of precision, recall, and F-score corresponding to each class (road, agricultural zone, urban zone, and vegetation zone) for Image 1 and 2 are computed. Furthermore, the final performance values for all images are shown. In the second experiment, the number of superpixels is calculated by means Eq. 2:

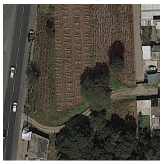
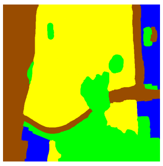
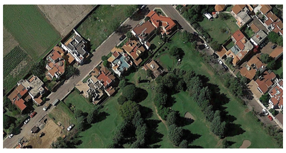
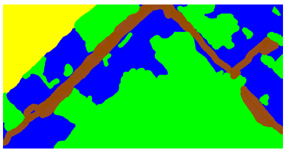
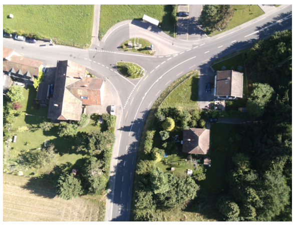
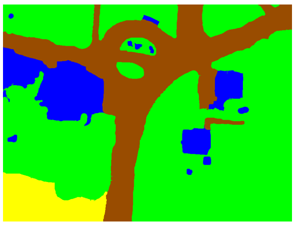
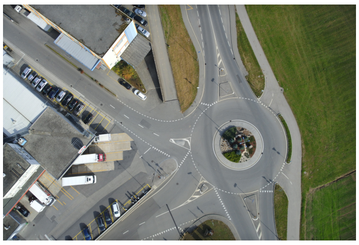
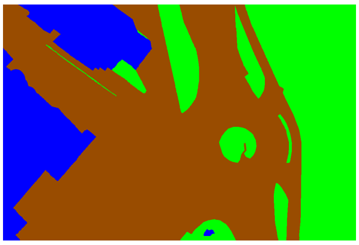
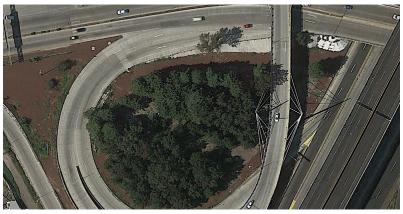
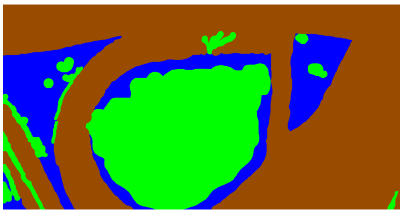
$$no\_spx = width \times height \times 0.003 \qquad (2)$$



(a) (b) (c)

Figure 3: (a) Original image. (b) Image partitioning into superpixels. (c) Image partitioning into superregions.

Table 1: RGB aerial image test, their size and their ground truth. The colours employed in ground true are paths in brown, green zone in green, urban zone in blue and agricultural zone in yellow.

Id / Size	**Image**	**Ground Truth**
Image 1 600×600		
Image 2 900×463		
Image 3 900×675		
Image 4 1100×733		
Image 5 1651×850		

where width and height correspond to the size of each image, the constant value, 0.003, was defined by a set of experiments. Finally, the third experiment, a superregion is composed of adjacent superpixels whose D_c is less than 30 (see section 3.4). The aim to group a superregion is to take advantage of the colour similarity among a set of adjacent superpixels, and only a subset of a randomly chosen superpixels was classified. For the last experiment, we consider two sizes of subsets, which were a) a quarter, and b) the half of the total superpixels of a superregion. To define the class of a superregion, four counters were considered (one for each class), which were initialized to zero and were increased according to the class assigned by the CNN model. When the evaluation of the subset of superpixels was finished, the class of the superregion was related to the highest counter.

In Table 2, the number of elements classified by our model are listed, where the number of elements is the highest for the first experiment, meanwhile the lowest number corresponds to the superregion-based with a quarter of elements. Corresponding to superpixel-based and superregion-based with half of the elements, there is no substantial saving in the elements classified. Concerning to the time required (in seconds) for preprocessing and to segment an aerial image is presented in Table 3. Wherein the behaviour is similar to the number of elements classifies, so that to high number of elements, the time required is high (pixel-based classification) meanwhile to lowest number of elements, the time required is low (superregion-based classification evaluating a quarter of superpixels). The difference between superpixel-based and superregion-based classification is between 2 and 3 seconds. On the other hand, the time required for grouping pixels in superregions is greater than the time to segment from Image 3 to Image 5, their final time are lower than pixel-based classification.

4.2 *Pixel-based classification*

In the first experiment, the number of elements classified was defined by the size of the image, that is, $weight \times height$. The performance measures for Image 1 and 2 for each class are shown in Table 4. Although our model was trained with another type of architecture of houses, it was able to classify correctly more than the half of pixels of Image 2 for the urban zone (F-score 0.762618). On the other hand, the worst F-score value was obtained for roads, 0.305186, which was less than 0.802189 (Image 1). The previous result can be generated because the patches corresponding to urban zone and roads in some cases have similar textures, in particular in those roof without waterproofing. The average performance values are presented in the first and the second rows of Table 5.

In Table 5 the average performance measure for each image are presented. According to the average of the precision (0.729796), the recall (0.736097) and the F-score (0.728108) values, our model is able to classify the most of pixels correctly. However, it mistook when the altitude of the image is near the 400 meters (Image 2) and the architecture of the houses is different to those employed to train the CNN model (Image 3).

4.3 *Superpixel-based classification*

In the second experiment, the number of elements classified was defined by Eq. 2. In Table 6, the performance measures for Image 1 and 2 for each class are presented. According to the F-score values, the use of superpixels improves the semantic segmentation for Image I, however the F-score values for agricultural zone and urban zone are worst than pixel-based classification approach (see Table 4). The average performance values (bold text) are presented in the first and the second rows of Table 6.

The average performance measure for each image using superpixel-based classification are exposed in Table 6. In accordance with the performance values, this approach improves the accuracy for Image 1, 3 and 4, due to the values are higher than those reported in Table 4. Additionally, the average values for this experiment has been improved.

4.4 *Superregion-based classification*

Due to the random selection of superpixels to define the class for a superregion, in this experiment ten runs were executed, therefore the statistical values (B: best, W: worst, Av: average, Md: median and SD: standard deviation) for precision, recall and F-score measures are presented in Tables 8 and 9. In Tables 10 and 11 the median values for ten runs are shown.

In Table 8, the statistical results for ten runs evaluating half of superpixels for a superregion is presented. For Image 1, the performance is better than the results to Image 2. With respect to the statistical results for ten runs evaluating a quarter of superpixels for a superregion (see Table 9), the performance values are lower than the first, however this semantic segmentation reduce the number of superpixels evaluated, therefore this is more fast.

In Tables 10 and 11 the average median values are presented for half and a quarter of superpixels, respectively. According to these values, the evaluation of half of superpixels reported the best performance measures, meanwhile the evaluation of quarter of superpixels had a similar behaviour than superpixel-based classification.

4.5 *Graphical comparison*

In Figure 4 the output images corresponding 1 and 2 are presented. Concerning to output Image 1 (middle column), pixel-based classification presents the salt-and-paper effect in the boundary of each region, mainly. Moreover, the regions labeled by a number represent misclassified regions because this are shadows. The regions surrounded by red rectangle represent an area of image with low accuracy of prediction and regions marked off pink rectangles misclassified some pixels. Superpixel-based classification and

Table 2: Number of elements classified by our model for each experiment.

Image	Pixel	Superpixel	Superregion	
			Quarter	Half
Image 1	360000	1069	460	960
Image 2	416700	1230	616	1062
Image 3	607500	1769	805	1581
Image 4	806300	2451	720	2020
Image 5	1403350	4127	1338	3722
Total	**3593850**	**10646**	**3939**	**9345**

Table 3: Time in seconds required for preprocessing and to segment each image for each experiment.

Image	Preprocessing		Segment			
	Superpixel	Superregion	Pixel	Superpixel	Superregion	
					Quarter	Half
Image 1	0.4971	3.1863	1448.0335	9.7861	3.8260	4.9720
Image 2	0.5756	3.8260	1656.0166	11.2636	5.0356	6.0649
Image 3	0.8152	7.4259	2678.378	17.5957	7.3329	9.1378
Image 4	1.0747	12.9124	3543.2988	26.6574	10.0399	12.9377
Image 5	1.8757	33.5103	5641.1023	55.4430	21.1628	26.9086

Table 4: Precision, Recall and F-score values for Image 1 and Image 2 using Pixel-based classification. Values in boldface indicate the average value.

	Image 1			Image 2		
	Precision	Recall	F-score	Precision	Recall	F-score
Roads	0.889540	0.730460	0.802189	0.286645	0.32629	0.305186
Agr Z	0.912811	0.800453	0.852948	0.561734	0.576946	0.569239
Urb Z	0.628760	0.766913	0.690999	0.694606	0.845396	0.762618
Veg Z	0.658451	0.882978	0.754362	0.84423	0.843212	0.843721
Avg.	**0.772390**	**0.795201**	**0.775124**	**0.596804**	**0.647961**	**0.620191**

Table 5: Precision, Recall and F-score values for each image using Pixel-based classification. Values in boldface indicate the average value.

	Precision	Recall	F-score
Image 1	0.772390	0.795201	0.775124
Image 2	0.596804	0.647961	0.620191
Image 3	0.66972	0.715261	0.684503
Image 4	0.831632	0.780933	0.80491
Image 5	0.778437	0.741131	0.755813
Avg.	**0.729796**	**0.736097**	**0.728108**

Table 6: Precision, Recall and F-score values for Image 1 and Image 2 using Superpixel-based classification. Values in boldface indicate the average value.

	Image 1			Image 2		
	Precision	Recall	F-score	Precision	Recall	F-score
Roads	0.914880	0.737076	0.816409	0.223892	0.250646	0.236515
Agr Z	0.913373	0.815941	0.861912	0.605537	0.55185	0.577449
Urb Z	0.664871	0.838919	0.741823	0.673891	0.846333	0.750332
Veg Z	0.674830	0.881962	0.764616	0.845352	0.847048	0.846199
Avg.	**0.791988**	**0.818474**	**0.796190**	**0.587168**	**0.623969**	**0.602623**

Table 7: Precision, Recall and F-score values for each image using Superpixel-based classification. Values in boldface indicate the average value.

	Precision	Recall	F-score
Image 1	0.791988	0.818474	0.796190
Image 2	0.587168	0.623969	0.602623
Image 3	0.674413	0.719813	0.688764
Image 4	0.834022	0.785795	0.808782
Image 5	0.775443	0.739259	0.753459
Avg.	**0.732606**	**0.737462**	**0.729963**

Table 8: Statistical results (B:best, W:worst, Av:average, Md: median and SD: standard deviation) for ten test for Image 1 and Image 2 using Superregion-based classification and evaluating half of the superpixels belonging to a superregion. Values in boldface represent average values.

	Image 1			Image 2		
	Precision	Recall	F-score	Precision	Recall	F-score
B	0.799210	0.821027	0.800211	0.594172	0.626659	0.603561
W	0.780585	0.799252	0.781955	0.558583	0.595353	0.575089
Av	0.789611	0.812634	0.791926	0.580797	0.614114	0.593997
Md	**0.79113**	**0.81579**	**0.79400**	**0.58189**	**0.616448**	**0.59545**
SD	0.006109	0.007718	0.006293	0.010209	0.011323	0.009934

Table 9: Statistical results (B:best, W:worst, Av:average, Md: median and SD: standard deviation) for ten runs for Image 1 and Image 2 using Superregion-based classification and evaluating a quarter of superpixels belonging to a superregion. Values in boldface indicate average values.

	Image 1			Image 2		
	Precision	Recall	F-score	Precision	Recall	F-score
B	0.777467	0.812371	0.782793	0.62784	0.71738	0.66644
W	0.734272	0.755387	0.724421	0.54104	0.58519	0.55893
Av	0.761531	0.791289	0.762080	0.57714	0.61632	0.59320
Md	**0.76166**	**0.79220**	**0.76482**	**0.57277**	**0.60958**	**0.58964**
SD	0.013380	0.016181	0.016911	0.02450	0.03704	0.02854

Table 10: Precision, Recall and F-score values for each image using Superregion-based classification. Values in boldface indicate the average value.

	Precision	Recall	F-score
Image 1	0.79113	0.81579	0.79400
Image 2	0.58189	0.616448	0.59545
Image 3	0.684791	0.753373	0.709578
Image 4	0.852012	0.777926	0.812681
Image 5	0.822926	0.788309	0.804562
Avg.	**0.746549**	**0.750369**	**0.743254**

superregion-based classification improved the misclassification into pink areas. With respect to regions labeled with the number 1 and 2 were improved. Finally those regions delimited by red rectangle preserves the behavior, due to any approach and our CNN model was not able to improve the misclassification. With respect to Image 2 (right column), the CNN model misclassified the roads, which were labeled as urban region. Furthermore, this problems happen in less degree with agricultural and vegetation zone. In these images, the plantations grow land share features with vegetation area, for this reason is classified as vegetation area.

4.6 *Discussion*

The accuracy performance of our CNN model approach can segment the most of image correctly, the average F-score for each approach classification is 0.732 approximately. This is better than [10], which is 55.52%, on the experiment

Table 11: Precision, Recall and F-score values for each image using Superregion-based classification. Values in boldface indicate the average value.

	Precision	Recall	F-score
Image 1	0.76166	0.79220	0.76482
Image 2	0.57277	0.60958	0.58964
Image 3	0.664193	0.722956	0.679302
Image 4	0.845415	0.777604	0.809196
Image 5	0.812956	0.788685	0.798829
Avg.	**0.731398**	**0.738205**	**0.728357**

where RGB satellite images were employed. Furthermore, this value (0.732) is greater than the overall accuracy (60-70 percent) reported by [8]. The use of superpixels allow defining more precisely the borders between classes. With respect the use of superregions, this reduce the time required to segment an image, but the final semantic segmentation depend on the randomly chosen superpixels. Two factors can affect the performance of the CNN model, which are the height of the image and the shadows presented, due to several elements are misclassified.

5 CONCLUSIONS

The focus of this work lies on testing the performance of a CNN model trained with RGB aerial images, that is without additional information such as DMS or NDVI. Moreover, the altitude of the images is closer than satellite images, that is, images taken from 100-300 meters above the ground, which represent a challenging task because these images contain more details than satellite images. Due to its high pixel resolution, the computing effort to process an image is reduced, in the number of elements to be classified, by using superregions or superpixels. According to the results, our model can predict most of the pixels correctly. Note that our pixel-based approach produces the salt-and-pepper effect, which is decreased with the use of superpixels and superregions. Although the accuracy is similar in the three approaches presented in this work, the use of superpixels and superregions reduce significantly the required time to segment an image.

In future work, we aim at optimising our proposed methods to enable on-line processing. Our goal is to carry out the classification during operation flight, in particular, by using small drones.

ACKNOWLEDGEMENT(S)

Thanks to Magda-Nayeli González-Sánchez for her support to build the dataset of this work. And, all authors acknowledge support from INEGI and CONACyT through project No. 268528.

REFERENCES

[1] Yaning Yi, Zhijie Zhang, Wanchang Zhang, Chuanrong Zhang, Weidong Li, and Tian Zhao. Semantic segmentation of urban buildings from vhr remote sensing imagery using a deep convolutional neural network. *Remote Sensing*, 11(15):1774, Jul 2019.

[2] Y. Babykalpana and K. ThanushKodi. Classification of lulc change detection using remotely sensed data for coimbatore city, tamilnadu, india. *Journal of Computing*, 2(5):46–56, 05 2010.

[3] Robail Yasrab. Ecru: An encoder-decoder based convolution neural network (cnn) for road-scene understanding. *Journal of Imaging*, 4(10):116, Oct 2018.

[4] Matija Radovic, Offei Adarkwa, and Qiaosong Wang. Object recognition in aerial images using convolutional neural networks. *Journal of Imaging*, 3(2):21, Jun 2017.

[5] Guangming Wu, Xiaowei Shao, Zhiling Guo, Qi Chen, Wei Yuan, Xiaodan Shi, Yongwei Xu, and Ryosuke Shibasaki. Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. *Remote Sensing*, 10(3):407, Mar 2018.

[6] Pascal Kaiser, Jan Wegner, Aurelien Lucchi, Martin Jaggi, Thomas Hofmann, and Konrad Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 07 2017.

[7] Maciej Dzieszko, Piotr Dzieszko, Slawomir Królewicz, and Jercy Cierniewskski. Digital aerial images land cover classification based on vegetation indices. *Quaestiones Geographicae*, 31(3):5–23, October 2012.

[8] Ming-Jer Huang, Shiahn-wern Shyue, Liang-Hwei Lee, and Chih-Chung Kao. A knowledge-based approach to urban feature classification using aerial imagery with lidar data. *Photogrammetric Engineering & Remote Sensing*, 74:1473–1485, 12 2008.

[9] Martin Längkvist, Andrey Kiselev, Marjan Alirezaie, and Amy Loutfi. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing*, 8(4):329, Apr 2016.

[10] Sakrapee Paisitkriangkrai, Jamie Sherrah, Pranam Janney, and Anton van den Hengel. Semantic labeling of

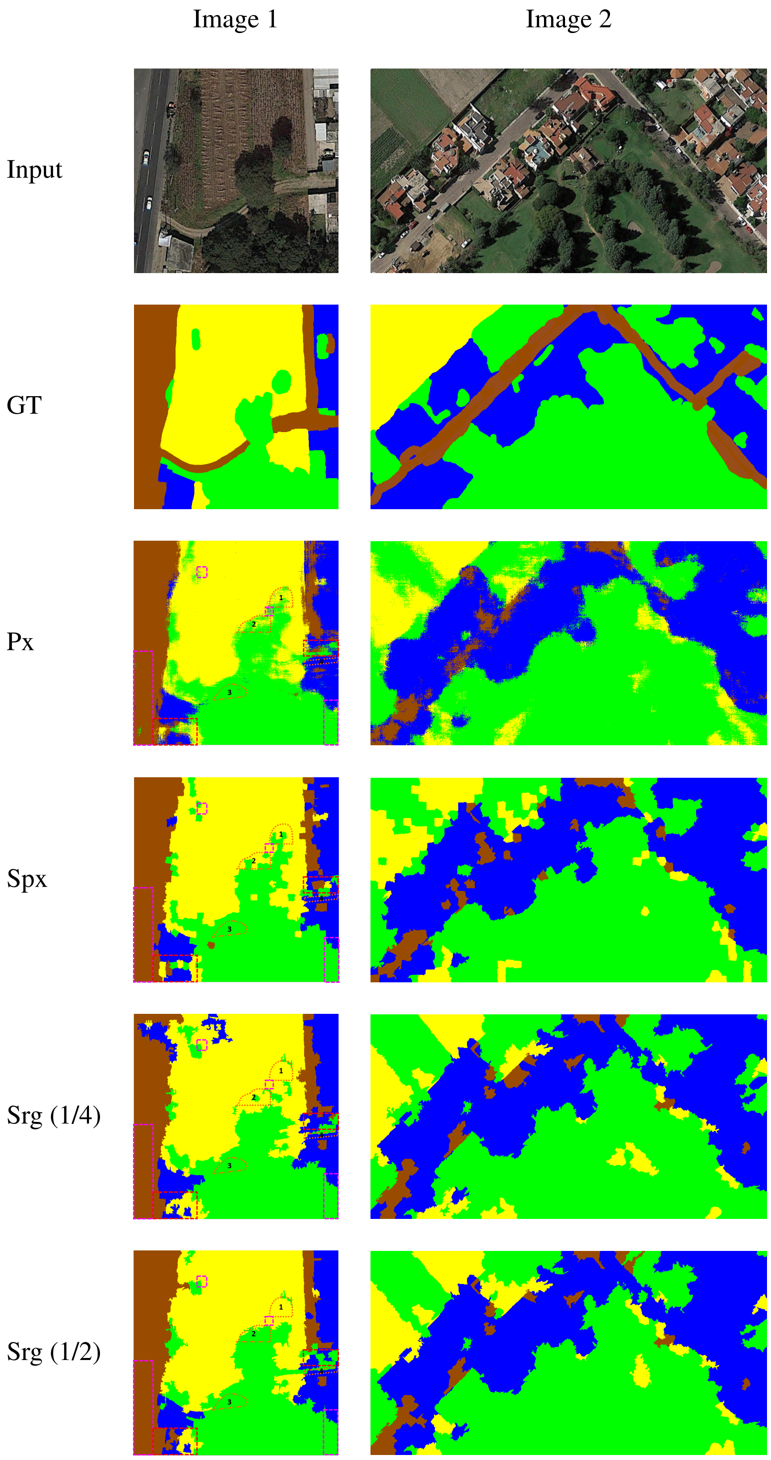


Figure 4: Output for each experiment: Px - pixel-based classification, Spx - superpixel-based classification, Srg 1/4-superregion-based classification 1/4, Srg 1/2 - superregion-based classification 1/2.

aerial and satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(7):2868–2881, 2016.

[11] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision (ACCV16)*, pages 180–196, Taipei, Taiwan, 2016.

[12] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[13] Chun Yang, Franz Rottensteiner, and Christian Heipke. Classification of land cover and land use based on convolutional neural networks. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-3:251–258, 04 2018.

[14] Lei Ding, Hao Tang, and Lorenzo Bruzzone. Lanet: Local attention embedding to improve the semantic segmentation of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–10, 2020.

[15] Jiankun Li, Wenrui Ding, Hongguang Li, and Chunlei Liu. Semantic segmentation for high-resolution aerial imagery using multi-skip network and markov random fields. In *2017 IEEE International Conference on Unmanned Systems (ICUS)*, pages 12–17, 2017.

[16] Dimitris Marmanis, Jermaine Wegner, Silvano Galliani, Konrad Schindler, Mihai Datcu, and Uwe Stilla. Semantic segmentation of aerial images with an ensamble of cnns. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3:473–480, 06 2016.

[17] Kai Yue, Lei Yang, Ruirui Li, Wei Hu, Fan Zhang, and Wei Li. Treesegnet: Adaptive tree cnns for subdecimeter aerial image segmentation. *Computer Vision and Pattern Recognition*, 2018.

[18] Vikas Gurumurthy. Encoder-decoder based cnn and fully connected crfs for remote sensed image segmentation. 10 2019.

[19] Jamie Sherrah. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *CoRR*, abs/1606.02585, 2016.

[20] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, page 1–14, 2015.

[22] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, Jan 2018.

[23] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *Technical report, EPFL*, 06 2010.

[24] Imanol Luengo, Mark Basham, and Andrew French. Smurfs: Superpixels from multi-scale refinement of super-regions. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 4.1–4.12. BMVA Press, September 2016.