Design of a Recognition System Automatic Vehicle License Plate through a Convolution Neural Network

P. Rajendra
Department of Mathematics, Madanapalle Institute of Technology and Science, Madanapalle, India

K. Sudheer Kumar
Department of Computer Science & Engineering, KG Reddy College of Engineering & Technology, Hyderabad, India

Rahul Boadh
School of Basic and Applied Sciences, K.R. Mangalam University, Gurgaon, India

ABSTRACT
The present work is a study on the practical application of Learning process (Deep Learning) in the development of a system of Automatic recognition of vehicle license plates. These systems commonly referred to as ALPR (Automatic License Plate Recognition) - are able to recognize the content of vehicles from the images captured by a camera. The system proposed in this work is based on an image classifier developed through supervised learning techniques with convolution neural network. These networks are one of the most profound learning architectures and are specifically designed to solve artificial vision, such as pattern recognition and classification of images. This paper also examines basic processing techniques and Image segmentation - such as smoothing filters, contour detection - necessary for the proposed system to be able to extract the contents of the license plates for further analysis and classification. This paper demonstrates the feasibility of an ALPR system based on a convolution neural network, noting the critical importance it has to design a network architecture and training data set appropriate to the problem to be solved.

General Terms
Deep Learning, Tensor flow, Python

Keywords
Convolution Neural Network, Deep Learning, ALPR

1. INTRODUCTION
1.1 Artificial vision systems
Artificial vision is a branch of artificial intelligence whose purpose is design computer systems capable of "understanding" the elements and characteristics of a scene or image of the real world. These systems allow extracting information - numerical and symbolic - from of the recognition of objects and structures present in the image. Artificial vision is closely related to Image processing and pattern recognition. In earlier, used to facilitate the localization and detection of areas of interest in the images; latter are used to identify and classify objects and structures detected according to their characteristics.

1.2 ALPR systems
Systems ANPR *(Automatic License Plate Recognition)* vehicles are a particular case of systems of artificial vision. These systems are designed to *"read"* the content vehicle registrations from images captured by a camera, and are mainly used in monitoring and control devices. In these systems the recognition of the images of numbers and letters can be implemented through different Machine learning, with the most common being the neural networks of type *Multilayer Perceptron* (MLP) and S*upport Vector Machines* (SVM). Even some systems choose to use commercial Optical Character Recognition (OCR) for this purpose.

1.3 Classification of images
The recognition or classification of images consists of assigning an image a label of a defined set of categories based on their characteristics. Despite seeming to be a relatively trivial problem from our perspective, is one of the most important challenges artificial vision systems. Factors such as scale, lighting, deformations or partial concealment of objects make the classification of images is a complex task, to which a great deal of effort to develop sophisticated pattern recognition techniques, which do not always produce the expected results. From the point of view of machine learning, the classification of images is a **supervised learning** problem, in which the classifiers algorithms generate a *model* from a *dataset* or set of previously categorized images. The model obtained is used later to classify new images.

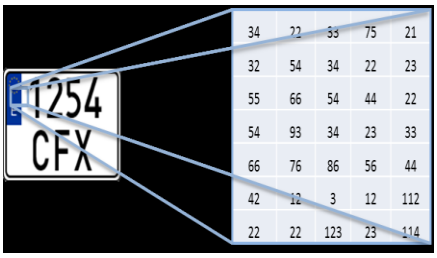

Fig.1: Digital image expressed as an array (pixel values are Fictitious)

For these algorithms the images are three - dimensional **matrices** whose dimensions are the width, height and color depth. The content is being of each position of the matrix a numerical value representing the intensity of each pixel in the digital image.

Until the popularization of convolution neural networks (CNN), the systems classifiers based on support vector

machines **(SVM)** were the ones that presented the best results in recognition problems and classification of images. The main drawback of these systems is that they require a prior process of **extracting the** relevant **features** of the Images, almost always designed to the problem of the intended purpose solve, for which sophisticated techniques of detection of objects are used and patterns - histograms of oriented gradients *(HOG),SIFT* descriptors etc. The SVM classifier algorithm is trained from the characteristics drawn to a subset of the training *dataset*, so the efficiency of the generated model depends on how representative they are. The most negative aspect of these systems is that they really **do not learn** the characteristics or attributes of each category, as they predefined in the extraction step. In addition, these systems are extremely sensitive to variations in scale, lighting, perspective etc.

1.4 Artificial neural networks

Artificial neural networks are an automatic learning paradigm inspired by the functioning of the biological brain. These networks are composed of interconnected neurons that collaborate to produce an output from the input data of the network. Each artificial neuron or *perceptron* is a processing unit receives a series of input signals multiplying by a given weight *(Synaptic weights).* The neuron calculates the sum of the product of each input by its corresponding weight - to which a correction factor is usually added or *bias* - and applies the resulting value to an activation function that produces a output value or other, depending on whether the sum of signals and weights exceeds one certain *threshold.*

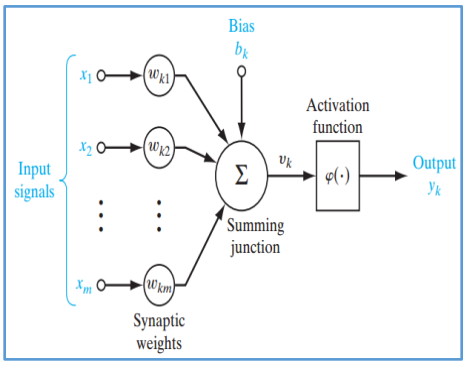

Fig.2: Schematic an artificial neuron with 'm' inputs

Artificial neural networks are organized in **layers** of neurons where each layer processes the information received from the previous layer. The number of layers and the type of neuron activation function determines the complexity of the problems that the network can solve: from detecting simple patterns in linearly separable data (one layer); to complex nonlinear relationships between the input data (more than three layers). Networks with one or more intermediate layers between the input and the output are which they are called **deep networks,** and are the basis of *Deep Learning*. In the following diagram you can see a network with a Hidden layer, it is also observed how in these networks the neurons of a layer are fully interconnected with those of the next. The great advantage of neural networks over other methods is their **machine learning** mechanism, through which it is not necessary develop selection processes and attribute extraction. The learning algorithm of these networks allows extracting the attributes or characteristics of each class from a set of training data previously classified. These attributes are the weights of the different neurons in the network, and their values are calculated iteratively by a supervised learning method called ***back propagation*** or *"Backward error propagation".*

In broad outline, the algorithm consists of two repeating stages iteratively for each element of the training set:
• In the first class to which it belongs is calculated input copy according to the current values of the network weights. Once classified, the algorithm determines the validity of such classification by an **error function** that calculates cost or how good or bad it is, comparing it with the class to which the example of training introduced into the network.
• Known error, the second stage of the algorithm propagates back to all neurons in the network that have contributed to the classification of example, each receiving the "portion" of the corresponding error in function of their contribution, so that they update the weights proportionally, so that the new values reduce the classification error.

Gradient descent method allows calculating a minimum value of the error function is obtained. To calculate this value, the algorithm descends iteratively by the slope (Gradient) from the current value, forward in each iteration step with a length or **rate** specific **learning.** The main drawback of the neural networks described in this section is the difficulties posed by the total interconnection between layers of neurons with increasing *dimensionality* of the data of entry. For example: a color image of 300x300 pixels requires a input layer with 270,000 pesos (300x300x3), if multiple layers are added intermediate network to the number of weights and *bias* necessary to grow exorbitantly, increasing the computational cost and the risk of over network training *(over fitting).*

1.5 Convolution neural networks

The **convolution neural network** (CNN) is a particular type of neural network. Neuronal system inspired by the functioning of the visual cortex of the brain. These networks are designed to solve artificial vision problems such as recognition of patterns, although they may have other uses such as classification of texts or processing of natural language. Like the "conventional" neural networks, they receive an input that transform through a series of layers of neurons, but in this case the input is an image represented as a three-dimensional matrix - Width, height and color depth - containing the numerical values of the Pixels. The layers of neurons are also organized in a way three-dimensional, and the result of the different transformations carried out on the net is the *class* or category to which that image belongs. The convolution networks are constructed using **four types of layers**: input layer, convolution layer, layer *pooling* and fully connected (FC, *Fully Connected).* **Convolution layers** in neurons are not fully interconnected, but rather with a small region of the anterior layer. In addition, this layer neurons share the same weights and *bias*, which greatly reduces the number of network parameters. Graphically, these clusters can be interpreted as a **window** that runs the

image or input layer, sliding from left to right and up and down until you reach the end. The **size of the sale** and its **length displacement** are parameters to be determined according to the characteristics of the problem. As the window moves, the corresponding neurons in the convolution layer calculate the product between the matrix of shared weights, *bias* and the pixel values of the region to which are connected, sending the result of this operation to an **activation function** determined - usually *RELU(Rectified Linear Unit)* function, which returns the *max (0, x)*value. The result of these calculations is a series of **maps activation** allow detecting the presence of a certain characteristic in the image of entry. This feature is defined by the weight matrix shared and the value of *bias;* the combination of both is called **filter** or *kernel* and its application to a region of pixels is called *convolution.* As you can see in the previous image, in the convolution layers it is several filters - weight matrices - to detect more of a feature in the picture. The convolution layers are usually accompanied by a **layer of *pooling*** which condenses the information collected, leaving only the maximum values of an area of the convolution layer, which dimensions of the input volume for the next layer. The typical **architecture** of a CNN network is a succession of layers *pooling* and convolution, being habitual that the last layer is a *Fully Connected* network which is responsible for calculating the scores obtained by the input image for each of the classes or categories defined in the problem. The **learning algorithm** of these networks is seen in *back propagation* gradient descent and other similar methods. But in these networks training aims at the different layers of neurons learn the filters or characteristics of low and high level (lines, edges, etc.) which represent each class or category of problem image. CNN networks are today the *"state of the art"* in the field of vision Artificial, where they have demonstrated a superior efficiency to other techniques, and this is due to three key **advantages** of the architecture:

- Ability to detect an attribute in any position of the image by sliding window.
- Tolerance to slight variations in rotation, translation or scale, thanks to the *pooling* layers.
- Ability to recognize complex attributes or characteristics of high level by combining the filters of the convolution layers.

An example of the predominance of convolution networks is the results of *ImageNet* the prestigious testing [1]. In these annual tests they complete teams of experts in artificial vision of the whole planet presenting systems that attempt to classify images into 200 predefined categories, starting from a training set consisting of more than 450,000 images. Since the year 2012 to date, the systems based on CNN networks that obtain the best results, with error rates considerably lower than those offered by other artificial intelligence techniques[2].

2. PROPOSED SYSTEM

The system **input** images are of vehicle registrations. In this paper, we used both real images such as computer generated, automatically scaled to a resolution approximately 230x50 pixels for easy processing. Once an image is entered, the system will execute the **segmentation** it is consisting detecting possible numbers and letters present on the license plate. The code module that performs this process leads to a segmentation based on the detection of contours, using it features the popular *OpenCV* computer vision library. The stage **character recognition** will be made through a CNN previously trained with a set of data with computer generated numbers and letters with different typographies and styles. The **output** of the system will be a string composed of the characters detected in the image of the entry plate.

3. DESCRIPTION OF THE DATA

3.1 Training data set

The effectiveness of a network-based image classification system CNN depends on both the architecture and network settings, and of the data set used for their training and must sufficiently broad and representative of the problem that is wanted solve. In order to train the neural network of the proposed system, the set *Chars74k* data created by T. M. Campos and Varma [3]. This *dataset* has over 74,000 images numbers and letters in PNG format organized in three collections: characters manuscripts, drawn characters and photographs of everyday scenes computer generated characters. This latest collection has more than 60,000 images in grayscale of digits and letters represented with different fonts and styles (normal, bold and italics), a priori suitable for the CNN network "learn" to recognize patterns of different numbers and letters of the license plate.

Fig.3: Sample images *dataset* Chars74

However, not all images in this collection are valid for the network training: lowercase letters and copies generated by "exotic" sources they are not representative of the problem. Excluding these images collection generated characters, computer is used in training to reduce to a total of 34,584 images of the digits 0-9 and letters AZ, mean about 950 images for each class of numbers and letters.

3.2 Data set to validate the system

Get a *dataset* with actual images of vehicle license plates is a more complicated task than it might seem, since in many laws this information is considered a character data personal. In the case of Spain, a legal report of the Spanish Agency Data Protection (AEPD) [4] concluded "the provisions of the Organic Law 15/1999 on Protection of Personal Data (Act) ". To avoid this problem has chosen to validate the ALPR system with a set of images of vehicle license plates mostly generated by computer, also including some real photograph taken with vehicle owner's consent. These "synthetic"

license plates have been obtained using tools Free Web sites *platesmania.com* [5] and *acme.com* [6].

The first allows you to create registrations under existing national format, two groups of characters consisting of a four-digit number, from the 0000 to 9999, and three letters, starting with the BBB letters and ending with the ZZZ letters, where the five vowels and the letters Ñ, Q, CH and LL are deleted. Furthermore, the tool website *acme.com* can generate state license plates according to different formats United States, but without any restrictions when entering combinations of letters and numbers.

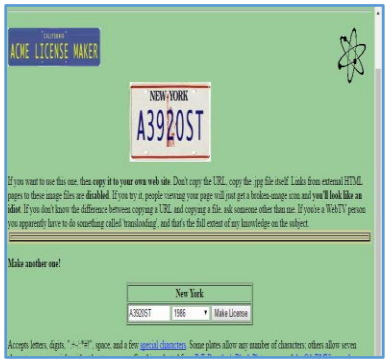

Fig.4: Generator license plates acme.com

Here are some examples of "synthetic" license plates are displayed generated with these tools to validate the system:

Fig.5: Examples of computer-generated license plates

4. TECHNOLOGIES EMPLOYED

4.1 Tools

The main tools used to develop the ALPR system have been the Python programming language and libraries *Tensor Flow* and *OpenCV* used to generate the convolution network and system pre image processing of enrollments respectively.

4.1.1 Python

Python is an interpreted programming language for general purpose, multiplatform and multi-paradigm (imperative, object-oriented and functional). It is a strongly typed language, i.e. you can not assign one varying a value of a different type to the original without an explicit conversion, being turn this typing *dynamic*: the type of the variables is determined in time execution depending on the assigned value. The reasons for the choice of this language are a clear and intuitive syntax. It facilitates code readability, and the large number of specialized bookstores available, such as *Numpy*, which allows a simple way all kinds of operations with vectors and matrices, fundamental when working with images and artificial neural networks.

4.1.2 OpenCV

OpenCV is a library of artificial vision developed by Intel and now released under BSD license, which has more than five hundred algorithms optimized to perform the main tasks of computer vision, such as image processing, feature detection or recognition objects. The library also has different learning algorithms machine as support vector machines (SVM), Naïve Bayes or KNN among others. It is written in C ++, multiplatform and has interfaces to work with languages like Java or Python. The large number of available algorithms, its execution speed and extensive user community that have, they make *OpenCV* an indispensable tool for developing systems artificial vision based on open source software.

4.1.3 Tensor Flow

It is an open source library developed by Google to facilitate design, construction and training systems deep learning. *Tensor Flow* is characterized by a programming model in which both the operation as data is stored internally in a structure graph, which facilitates the display of dependencies between operations and allocation to different devices such as graphics processors. While it is true that there are other libraries *Deep Learning* support Python that offer similar programming model; TensorFlow presents a fairly clear syntax and has a powerful tool display (*Tensor Board*) that allows us to understand, debug and optimize the graph computing offering all kinds of statistics. In addition, Tensor Flow behind the research team is project *Google Brain*, which has some of the leading experts in *Deep Learning* as Geoffrey Hinton and Alex Krizhevsky, authors of the first it based on a CNN network that won the test in 2012 system ImageNet [7].

4.2 IMAGE PROCESSING

As it was said in the system description, we first process running once introduced the image of a license plate It is the character segmentation, which aims to detect those regions image containing numbers and letters that subsequently try to identify the neural network classifier. The Python module that executes this process is based on *OpenCV* and employs the following image processing techniques:

4.2.1 Conversion to grayscale

The color information is not required for the problem posed and elimination facilitates the detection of the characters of the license plate. for this therefore, the first transformation is applied to the image is a conversion to grayscale, or what is the same, convert the image into a matrix in which each value represents the gray level of the corresponding pixel. This transformation is performed by the function *cvtColor* of *OpenCV*.

4.2.2 Gaussian Blur

The second transformation is done is to apply an image effect smoothed to remove noise. Specifically, a blur filter applied Gaussian mix consisting slightly gray levels between adjacent pixels, which results in an image with edges more

soft where small details are lost, similarly to what it occurs in blurred pictures. This transformation is performed by function *Gaussian Blur* of *OpenCV*.

4.2.3 Thresholding

The *thresholding* or "threshold technique" is a method that allows you to converting image in black and white setting a value from which all excess pixels that are transformed into a binary color (white or black), and rest counter. In the case of this vehicle registrations transformation creates images which clearly fall defined outlines of the characters, facilitating the process of segmentation or isolation areas containing them.

In the proposed system this transformation is performed by the function *adaptiveThreshold* OpenCV, wherein the threshold value is calculated for different regions of the image, providing good results even if there illumination variation image.

4.2.4 Edge Detection

The contours or edges are sets of curves joining adjacent pixels having the same color or intensity. These curves can locate the borders of the objects in the image, and its detection is essential for artificial vision system can recognize or detect shapes in a image. In the proposed system this detection is the last stage of pre-image processing of tuition, and is carried out by the functions *findContours* and *boundingRect* of *OpenCV*.

4.3 NEURAL NETWORK ARCHITECTURE

The main component of ALPR system presented in this work is its convolution neural network (CNN). This is responsible for classifying characters of the license plates from images taken in the module segmentation, for which a prior process of training is necessary for the network to learn the characteristics that define the different letters and numbers that may be present in an enrollment.

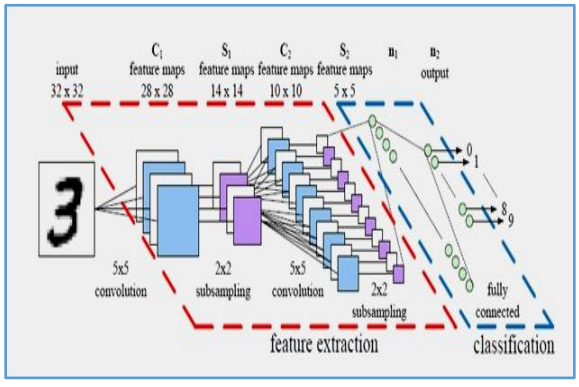

Fig.6: Architecture of the convolution neural network proposal

This architecture is based on the convolution network *LeNet-5* proposal by Yann Lecun [8] for recognition of digits, adapting the number of layers and values of various parameters to the needs of the problem raised in this paper. This design also follows the general strategy proposed by Simard et al [9] for visual analysis of documents by convolution networks, it consist extract simple features of the characters in the first network layers, and later turns them into complex features thanks to the combination of the various filters of successive layers convolution.

5. IMPLEMENTATION

5.1 Data processing

Then the various operations are described preprocessing they carried out in the system, both the segmentation module characters, such as during the loading process image set training the neural network.

5.1.1 Pre-processing input images

The various processing techniques described images that apply to enrollments coming into the system - conversion grayscale, Gaussian blur, *thresholding* and detection contours - in order to isolate those regions that may contain or letter of enrollment. Once identified these "regions of interest" (ROI), the Python module segmentation process manager performs a series of **checks** to determine if their content is effectively a digit or a letter that is to be analyzed and classified by the neural network. These checks are basically checked whether the height, width and pixel aspect ratio i.e. the ratio between width and height the region of interest - are within certain values. The minimum and maximum values of these parameters are determined by the proportions they should have the characteristics of a standard enrollment approximately 230x50 pixels, being the resolution to which you scale all the images that enter the system. If the proportions of a given region are valid, i.e. they are within limits, the next step is to extract its contents and save it as an image of 32x32 pixels that can be analyzed by the CNN network system.

5.1.2 Pre-processing training data set

To train the CNN network is necessary to import the subset of representative images of the problem that contains the *dataset Chars74K* in a data matrix. Each subset of the *dataset* is composed of images in grayscale computer generated characters corresponding to the numbers 0-9 and uppercase letters AZ. The big advantage of using a network neuronal as ranking algorithm is their learning ability automatic, thanks to which it is not necessary to develop extraction processes attribute or use techniques to reduce the dimensionality as PCA (*Principal Component Analysis*) or the like; greatly simplifying the pre-processing tasks of the training data.

5.2 Network training

The training process of the CNN network system is based on the combination algorithm *back propagation* with the drop method gradient. It is basically an optimization problem in which iteratively weights and - the values of the parameters are searched *bias* network – making the value of the error function or cost is minimal. For network CNN system, the selected function to measure how good or bad the classification of images is the error of **cross entropy** (*cross*

entropy Error), one of the usual design of convolution neural network. Of the different variants of gradient descent method has been chosen version **stochastic or incremental** , that is to calculate iteratively the gradient for a subset or *batch* of training data, updating the network weights by propagating errors to back, without waiting for the full data set is processed. From the standpoint of computational cost, this method is more efficient to process the gradient in each iteration the *dataset* completely, because usually provide faster convergence. Although it is also true that runs the risk of being "stuck" in a local minimum of the error function.

5.2.1 Initialization

Before running the training process is necessary to generate and initializes by *Tensor Flow* network architecture. Briefly, in the code module, A declares **graph operations** that contains both the variables that store data training, weight matrices, *bias* and outputs of the network; as calls the primitives *framework* to define the error function, the layers convolution (with window *stride*, filters and activation functions), layers *pooling*, layer FC, exit *softmax* and the call to update weights by *back propagation* and gradient descent. In the next picture you can see the graph of the CNN network system exactly it represents as the display tool *Tensor Board* included in the *framework*. Besides declare the structure of the network, in operation network indicated also how it should be initialized.

5.2.2 Training

Once declared and initialized the network, the next step is to import the set of training images preprocessed by the code module. Before starting the training, this set of images is divided randomly into three subsets called *training*, validation and *test*: the first contains a total of **28,012** images, the second**3113** and third **3459**. The training set is used to determine the weights of the network allow classification with the lowest possible error images, validation to check if over fitting (occurs *over fitting*) during training and test set to estimate the accuracy of the system to classifying images that have not been analyzed during the iterations of training. The section of code responsible for implementing the training process - makes a total of **4,500** iterations selecting each one lot or *batch* of **300** images of the training set to optimize network weights by function *Gradient Descent Optimizer* of *Tensor Flow* with a learning rate of **0.0001**; which means that the end of the will process each image is analyzed about **50 times** approximately.

5.3 Evaluation

To evaluate the effectiveness and efficiency of the learning process of the network has chosen to monitor the evolution of the overall accuracy and error in classifying images of subsets of *training* and *validation* during the iterative training process. Tracking accuracy detect if overload is occurring training (over *fitting*) network, i.e., if the weights and *bias* are excessively adjusting to the peculiarities of images. Training subset, which may cause once trained the network is not able to correctly classify the different images analyzed during learning. In the next picture you can see the evolution of **precision** in both the training set and the validation for 4500 iterations.

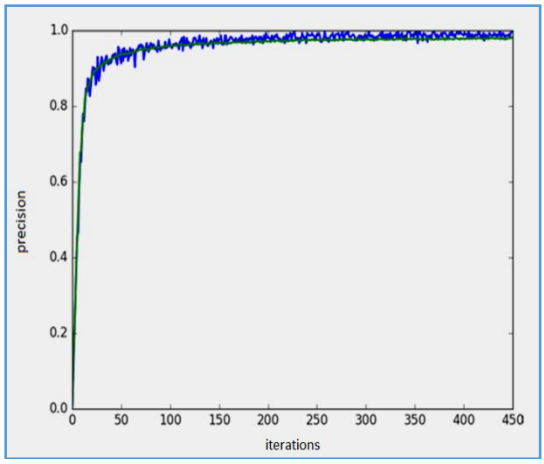

Fig.7: Evolution of precision during training

As you can see the distance between them is very small, which a priori is a good indicator that has not been produced *over fitting*. You can also observe the high accuracy obtained in both subsets: about 99% to 98% and training for validation. For the set of images of **test** accuracy obtained with the settings defined in the preceding paragraph is **97.83%**. Regarding the role of **errors**, the following chart shows its 4,500 iterations. Here you can see a fall very pronounced during the first 500 iterations and quite softer but constant for the rest, indicating that in principle the learning rate chosen for the gradient method seems appropriate.

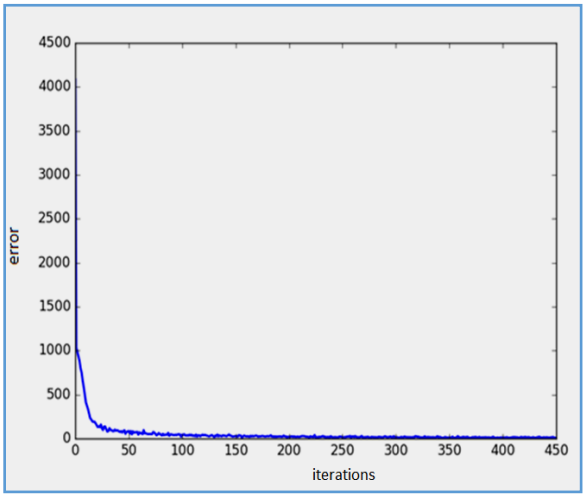

Fig.8: Evolution error function during training

In addition to tracking the above figures, the assessment effectiveness of the learning process is completed running the network trained with a sample of digits and letters extracted license plate images synthetic and real.

5.4 OPTIMIZATION

The characteristics of the network architecture (layers, filters, window size, etc.), as learning process parameters (number of iterations, batch size, learning rate, etc.); they

have been determined empirically in particularly following the procedure proposed by Nikhil Buduma [10]. It is adjusted iteratively parameters network architecture and training process function results. Apart from the above adjustment process diagram for optimization network have also followed some of the recommendations and techniques common in the design of convolution neural network [11], namely:

• Review training data set to delete images not representative of the problem. Here is the representations exotic or ambiguous character. To detect them, apart from manually review the *dataset*, it has been of great help assess the mistakes made by the network with sample images enrollment during training.

• Work in learning algorithms machine consists of subtracting the average data matrices each column. This focus usually comes accompanied by a normalization of data [13-16], in the case of images is not necessary since the values of pixels are on the same scale (0-255).

• *Dropout* of neurons fully connected layer (FC) is about fairly new technique to reduce overtraining. It was by Srivastava et al [12] in 2014 and it is "off" randomly some neurons during training, so only the weights of the remaining active are updated.

6. RESULTS OBTAINED

Once trained network CNN is just check the effectiveness of license plate recognition system designed. To this end they have been used three collections of images, two of them created with tools. Details of each are as follows:

1. Collection of 100 images of randomly generated by licensed plates computer according to the current national format, i.e. two groups characters consisting of a four-digit number, from 0000 to 9999, and three letters, where the five vowels and the letters Ñ, Q are deleted, CH and LL.
2. Collection of 100 images of randomly generated by licensed plates US computer according to different formats. Since there is no a single standard in this country, has chosen to use one of the common formats, consisting of two groups of three characters, the first with numbers 000 to 999, and the second with letters from AAA to ZZZ.

The results obtained for the three collections are:

Collection	**Total images**	**Correct hit**	**Failures hit**
National Matriculas	100	100	0
Matriculas USA	100	60	40

The results for the 100 images in **national format** are really spectacular, obtaining **100%** of correct recognition content of these images of license plates. In the case of **real images** the results are equally impressive **100%** hit although it is important to note that this sample is too small to be considered significant and also images they have been obtained on highly concessional terms.

7. CONCLUSIONS

The first conclusion is the confirmation of the feasibility of automatic license plate recognition (ALPR) based on a convolution neural network. The use of these networks creates a system really capable of "learning" the characteristics of the different digits and letters of the license plate, without complex mechanisms attribute extraction. The second is the realization of the importance of having a set of images of sufficiently extensive training and representative of the problem to obtain satisfactory results with a convolution neural network. In the case of the system presented in this work, the *dataset* used was not specific to the problem, and proof of this is the large number of images of characters in lowercase or exotic sources that had to be eliminated to optimize the learning process. The third conclusion is so laborious that can be optimized learning process convolution neural network, and poor about literature there currently. During the development of this paper has been necessary to repeat many times the proposed process to give architecture and process adjustments training offered reasonably satisfactory results.

We can extend this work by adding a pre-segmentation module to allow characters automatically locate vehicle registrations in photographs. Review the segmentation module to avoid the use of measures absolute in pixels to detect regions of interest. Expand the *dataset* training, gathering new images or by using techniques such as "data *augmentation* "to create new images from existing transformations, which undoubtedly improve the percentage of correct system.

8. REFERENCES

[1] ImageNet Project. http://image-net.org - March 2017.

[2] ImageNet ILSVRC2015.http://image-net.org/challenges/LSVRC/2015/results - March 2017.

[3] Chars74k dataset. TE Campos, BR Babu, M. Varma. http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/ - March 2017.

[4] Report 425/2006. Spanish Data Protection Agency (AGPD).

[5] Platesmania License Plate Generator. http://platesmania.com/es/informer - April 2017

[6] Acme License Maker. http://acme.com/licensemaker - April 2017

[7] ImageNet Classification With Deep Convolutional Neural Networks. Krizhevsky et al.http://papers.nips.cc/paper/4824-imagenet classification- with-deep-convolutional-neural-networks.pdf March 2017

[8] Gradient-Based Learning Applied to Document Recognition. LeCun et al.http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf - March 2017

[9] Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis.Simardetal.

http://research.microsoft.com/pubs/68920/icdar03.pdf - April 2017

[10] Fundamentals of Deep Learning. Nikhil Buduma. O'Reilly. http://shop.oreilly.com/product/0636920039709.do - April 2017

[11] Deep Learning Book. Ian Goodfellow, Yoshua Bengio,Aaron Courville. MIT Press.http://www.deeplearningbook.org - March 2017

[12] Dropout: A Simple Way to Prevent Neural Networks from Over fitting. Srivastava et al. https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf - May 2017

[13] Subba Rao, A et al. "Numerical Study of Non-Newtonian Polymeric Boundary Layer Flow And Heat Transfer from a Permeable Horizontal Isothermal Cylinder." *Frontiers in Heat and Mass Transfer FHMT)* 9, no. 1 (2017).

[14] Rajendra, P., et al. "Potential Distribution in a Nano Channel of an Electrolytic Solution by Finite Element Method." I J C T A, 9(32), 2016, pp. 145-151

[15] Padidhapu, Rajendra, Shahnaz Bathul, and V. Brahmajirao. "Least Square and Gauss Jordan Methods applied to fit the Dielectric data Vs. Concentration of Ionic Liquids." *International Journal of Science and Technology* 2.1 (2013).

[16] Boadh, Rahul, et al. "numerical simulation of boundary layer flow parameters by using wrf-arw model over a tropical region.", Jr. of Industrial Pollution Control 33(1) (2017) pp 1148-1154.