https://doi.org/10.31871/IJNTR.6.12.25
International Journal of New Technology and Research (IJNTR)
ISSN: 2454-4116, Volume-6, Issue-12 December 2020 Pages 118-121

High Accuracy Data Management and Processing Models

Mallioris Panagiotis, Sidiropoulos Athanasios, Bechtsis Dimitrios, Vlachos Dimitrios

***Abstract*— Data management and processing methods are used for efficiently handling time consuming and critical activities. With the up rise of digital technologies, data processing is used for supporting operational processes in everyday activities at a 24/7 perspective. Data sources vary from simple numerical data and text data, to images and videos that should be efficiently analyzed in order to be used at the decision-making process. The proposed research work focuses on the implementation of data processing algorithms for high accuracy license plate recognition, that could be used in Digital Supply Chains. Open source frameworks and tools (TensorFlow and Keras API) were used, in order to implement neural network methods and examine their accuracy. After a brief introduction, several methods are presented with the use of ssd_mobilenet_v1_coco and ssd_mobilenet_v1_ quantized TensorFlow models. The final model successfully recognizes license plates with high accuracy.**

***Index Terms*— plate recognition, neural network, categorical cross entropy, TensorFlow, Keras**

I. Introduction

The increasing demand for artificial intelligence (AI) applications establishes neural network as a state-of-the-art topic. Speech Recognition, computer vision and image processing are some of the many real-world applications where neural models are extensively used. In this paper, the basic characteristics of different types of neural networks are presented in order to provide a basis for the license plate detection model. The license plate recognition model is extensively described, the networks parameters are explained, and the results of our model are presented.

II. Literature Review

Neural networks (NNs) are a modeling tool for non-linear statistical dataconsisting of interconnected nodes which model complicatedcorrelations between inputs and outputs [1]. The connections of the neurons are presented as weighted links; a positive weight indicates an excitatory connection, while negative values demonstrates inhibitory connections. Every weight influences the input by providing a certain output with the help of an activation function. Neural Networks can be categorized as follows: (a) linear regression, (b) binary classification and (c) multiclass classification neural networks. The above-mentioned categories are analyzed in the following sections.

A. Linear Regression

Linear regression isfitting a linear equation to observed datain order to model the relationship between two continuous variables, with the first variable considered the explanatory variable and the other the dependent variable [2]. A neural network is trained to predict the output of a specified input using linear regression, by minimizing the loss in a dataset using the mean square error (MSE) of each point. The within-sample MSE of the predictor is computed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 (1),$$

where n is the data points on all variables of the sample, Y is the vector of observed values of the variable being predicted and Ŷ are the predicted values.

B. Binary Classification

Binary classification problems focus on measuring a series of attributes in order toassign the output of the model to one of two categories [3]. The commonly used level values in a binary classification problem, are 0 and 1. However, in a binary classification problem with neural networks the output can be a float number (between 0 and 1) and this indicates the probability of the classification. For example if a large dataset of human weights and ages is used and a neural network has to decide whether a person of a certain age and weight is obese the output will be most likely a float number between 0 and 1 that indicates the probability of the obesity behavior.

It is important in such cases to properly define the activation function employed by the neural network. One of the most common functions for classifying binary datasets is the sigmoid activation function. The sigmoid function has a range of values between 0 and 1 with a certain probability. When x is approaching negative infinity, the output is closer to 0 (almost 0), in the other side, when x is approaching positive infinity, the output is closer to 1 (almost 1).

The sigmoid function could be presented as an equation, which converts the variable x to a probability:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1} (2)$$

Another powerful transformation tool that is used in order to indicate a binary classification model is the Cross Entropy. Cross-entropy is a measure of the difference between two distribution functions within a neural network [4]. Low Cross Entropy values imply that an efficient neural network is constructed whereas high values present a non-efficient and

Mallioris Panagiotis, Industrial Engineering and Management Department, IHU, Sindos, Greece
Sidiropoulos Athanasios, Mechanical Engineering Department, AUTH, Greece
Bechtsis Dimitrios, Industrial Engineering and Management Department, IHU, Sindos, Greece
Vlachos Dimitrios, Mechanical Engineering Department, AUTH, Greece

less accurate network. The Binary Cross Entropy Equation is presented in Equation 3:

$$H_p(q)=-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i))+(1-y_i)\cdot\log(1-p(y_i))(3)$$

C. *Multiclass Classification*

Multiclass classification is used for classifying three or more classes in a dataset. The output of a Multiclass Classification is the probability of an item to be included in a specific class. For example, it is possible for a neural network to categorize and recognize various animals only by taking as input specific parameters like shape, size or color. Every neural network is a complex structure that can be used for processing a dataset using testing and training subsets. The first step is to collect a significant amount of unbiased data, like properties, sounds, pictures of all representative classes in order to create an unbiased dataset. In machine learning,biased datais when certain elements of thedatasetare more heavily weighted and/or represented than others, which can result in low accuracy and analytical errors. For example, if we want to create a model that predicts the weather in a certain place, we must have weather measurements for at least one year. If we take measurements only within winter's months the dataset will be biased, and our final model will have low prediction accuracy. The most used structure of a Multiclass Classification model is the Multilayer Perceptron (

Figure 1).

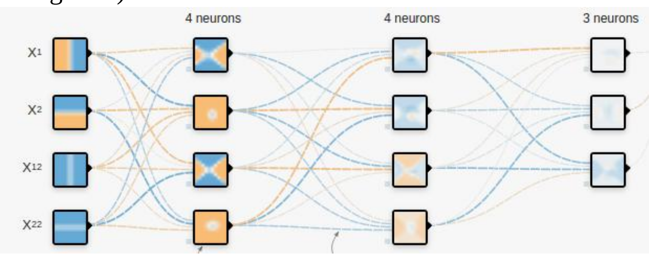

Figure 1. The architecture of Multilayer Perceptron contains multiple inputs (for ex. Weights, size, color) , various hidden layers , size of which could be defined by a user and can vary dependently of the problem faced, and one or more outputs.

The Inputs (x data) travel through the perceptron while constantly changing values by using the weights from the links among the neurons. Finally, the result is fed to the activation function and we get a feedback through back-propagation in order to reduce the loss. In order to minimize the error in every output the weights self-adjust depending on the difference between predicted outputs and training inputs. Therefore, there are two basic conditions Multilayer Perceptron should always contain, maintained relative magnitudes, and the probability outputs must add to one. The function that satisfies the aforementioned conditions is the Softmax activation function. The Softmax function is presented in Equation 4:

$$P(scorem) = e^m / \sum_{i=1}^{n} e^j \ (4)$$

This equation refers to the probability of a certain score m when there are n classes are in a dataset. Multiclass classification can be defined as accurate or not simply by calculating the cross-entropy function. Multilayer Perceptron utilizes categorical Cross entropy. The equation of categorical Cross entropy is as follows:

$$CCE=-\frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{J} y_j \cdot \log(\hat{y}_j)+(1-y_j)\cdot\log(1-\hat{y}_j)\ (5)$$

High complexity multi-layer perceptrons need relatively high computational power and as a result they have a relative slow response in multidimensional and complex datasets. They are commonly used on speech recognition or machine translation projects. Another major category of a Multiclass Classification model is the Convolution Neural Network. Convolution networks reduce the computational power even for demanding projects by utilizing additional convolution and pooling layers along with perceptron layers [5]. The size of the testing and training data will be relatively smaller when it reaches the perceptron layers after convolution and pooling takes place. They are commonly used on Image Processing and Computer vision projects that usually include multidimensional and complex datasets. The activation function that is commonly used on convolution neural networks is the ReLu function which will output the input unchanged if it is positive, otherwise, it will output zero. The advantage of this function is that it is easier to train and achieves better performance for more demanding projects. The form of the ReLu function is the following Equation 6:

$$R(z)= \max(0,z)(6)$$

A known convolution model is the leNet model. The leNet model takes as an input a random n-dimensional array which contains the intensity of pixels in an image. Through a series of convolution and pooling layers creates an output that memorize crucial parts of information for this image like edges or color intensity in a one-dimensional array. In the leNet-5 architecture the layer composition consists of 3 convolutional layers, 2 sub sampling layers and 2 fully connected layers.

III. Plate Recognition

The main objective of this publication is the creations of an effective and efficient model that uses neural networks capable of real time license plate recognition using a simple camera. As a first step, when facing a project with neural networks it is recommended to select the appropriate type of the neural network architecture. Plate recognition is a high demanding problem that requires a large set of objects as an input. A database of license plate pictures will be the input dataset. This can be characterized as a multidimensional and complex dataset. Moreover, it is important to create a model that can correctly identify plates among various objects (in the dataset we will include not only license plates, but other objects too) and it is critical to state that the dataset will also contains pictures with different angles in order to create an unbiased dataset. Considering all the above, a Multiclass Classification model was used that can handle highly demanding, multidimensional and computational complex datasets and specifically a Convolution neural network.

A. *TensorFlow and Keras Api*

The open source TensorFlow API was selected as a stable implementation of the neural network. TensorFlow is a library that enables the user to develop and train complex datasets and it is used in machine learning problems [6]. For this particular use case the TensorFlow 1.8 version was used alongside with Python 3.6. Keras is an open source software library for python which provides an interface for the neural networks. Furthermore, Keras is compatible and interoperates with the TensorFlow. Also, TensorFlow uses the Keras layers which are basic building blocks of neural networks that consist of an activation function and several

https://doi.org/10.31871/IJNTR.6.12.25
International Journal of New Technology and Research (IJNTR)
ISSN: 2454-4116, Volume-6, Issue-12 December 2020 Pages 118-121

filters (Dense). Moreover, Keras provides models which group layers in order to form and object with training and inference features.

B. Step by step model build

The first step for the implementation of the model was the creation of significantly large dataset of images that include license plates numbers in different angles and colors. 143 images were gathered either from internet resources or they were manually photographed by the authors. In order to make an effective and efficient model and after conducting the first experiments the dataset was split to the below two parts: (i) 10 % of the images were characterized as testing images; and (ii) 90% of the images were included to the training set. This is a prerequisite as the neural network utilizes a few images for testing and more images for training. The next step was to create a csv file for each category that contains all the critical information for each image like the label or the shape in order to be fed to our model.

Csv files enable TensorFlow to create a set of record files (train.record and test.record) using generate_tfrecord, which are used for the implementation of the model. TensorFlow provides a large amount of pre-build convolution models with different characteristics that could be used to a variety of projects. In the presented use case, we used two models: (i) the ssd_mobilnet_v1_coco model was used that has a high level of accuracy (COCO Map[^1] =21) and has a fast response time (30 ms); and (ii) a ssd_mobilnet_v1_quantized model with similar characteristics (COCO Map[^1] =18, 29ms) that was used as an alternative solution.

Finally, TensorFlow also provides a training python file that will automatically train the model which contains the record files and provide an output checkpoint file(ckpt) which includes all data until the last checkpoint of the last epoch that algorithm reached. On each epoch the model calculates the loss from the categorical cross-entropy for increasing the accuracy. In the end an inference graph will be exported from the last checkpoint which will be used on the webcam initialization python code.

```
Accumulating evaluation results...
DONE (t=0.04s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.703
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.839
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.821
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.406
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.850
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.609
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.752
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.783
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.633
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.887
```

Figure 2. License Plate ssd_mobilenet_v1_coco (b) custom model evaluation results (last checkpoint)

IV. Results

In order to compare the effectiveness and efficiency of the model, different classifications/categories of the same dataset were used, and different network architectures were created. The following experiments took place:

a) 5 images for testing, 50 for training using ssd_mobilenet_v1_coco

b) 19 images for testing, 124 for training using ssd_mobilenet_v1_coco

c) 19 images for testing, 124 for training using ssd_mobilenet_v1_quantized

The results captured when the TensorFlow had been reach to a checkpoint after running for 5000 steps. Regarding the coco model (a) the output presented in Figure 3 was not very efficient and accurate with a categorical loss of 1.35. The object detection algorithm struggled to identify an object from a plate and indicate the exact boundaries of the plates because it often made wrong decisions. The main fault was the small number of images that we gave as input dataset to our convolutional model for testing and training. After this experiment it was obvious that model needs more images in order to improve its accuracy (at least 100 - 500 unbiased images).

Figure 3 Plate recognition with the ssd_mobilenet_v1_coco model (a)

The second coco model (b) with categoricalloss of 0.35 was very efficient and provided fast and accurate results (Figure 4). The increase of testing and training images assisted the model to better indicate the boundaries of the license plates.

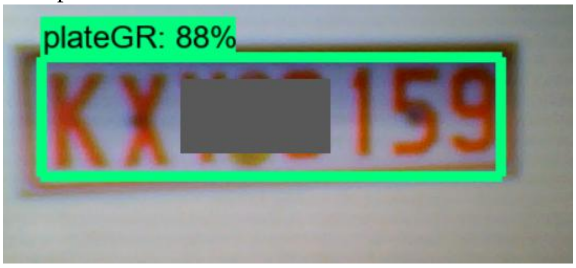

Figure 4 Plate recognition with the ssd_mobilenet_v1_coco model (b)

Finally, the third model c) had a similar output with the second one with a total categorical loss of 0.33. The main difference between the two models (b and c) is the size of the coco model file. In model b the system creates an inference graph of 22.247 KB, while in model c the quantized model creates an inference graph of 2.150 KB making it more accessible and suitable for Mobile App projects knowing that two models have same execution time. As it is presented in (Figure 5) the output was also efficient.

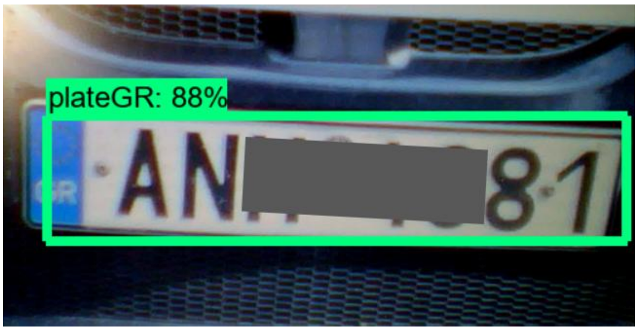

Figure 5 Plate recognition with the ssd_mobilenet_v1_quantized model (c)

V. CONCLUSIONS & FUTURE RESEARCH

The continuous increase on computing devices has as result unexploited processing power that could be used to solve real-world problems. Image processing and object recognition can be used in supply chains, in the automotive industry and in many aspects of our everyday lives. In this paper, alternative models for vehicle plate recognition are presented and evaluated. Two of them (models b and c) have high plate recognition accuracy in real-time employing the processing capability of common mobile devices. Many API's like TensorFlow, Keras, Yolo provide pre-build models helping beginners or more experienced users to implement efficient custom classification projects. A comparison between the presented models' accuracy with other models' accuracy would be useful to conclude to the most suitable solution for this certain application.

REFERENCES

[1]Jonathan Penm, Betty Chaar, Rebekah Moles, Jack Penm, 37 - ''Predicting ASX Health Care Stock Index Movements After the Recent Financal Crisis Using Patterned Neural Networks'', Editor(s): Carsten S. Wehn, Christian Hoppe, Greg N. Gregoriou, Rethinking Valuation and Pricing Models,Academic Press,2013

[2]Gangadhar Shobha, Shanta Rangaswamy, Chapter 8 - Machine Learning, Editor(s): Venkat N. Gudivada,C.R. Rao,''Handbook of Statistics'', Elsevier, Volume 38 , 2018

[3] *G. Parmigiani, in "InternationalEncyclopedia of the Social & Behavioral Sciences", Chapter 5.2 Binary classification, Editor(s) : Neil J. Smelser and Paul B. Baltes, 2001*

[4]Xiaowei Chen, Samarjit Kar, Dan A. Ralescu, "Cross-entropy measure of uncertain variables",Information Sciences,Volume 201,2012,

[5] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging ***9,*** *611–629 (2018).*

[6] Ladislav Rampasek, Anna Goldenberg,"TensorFlow: Biology's Gateway to Deep Learning?", Cell Systems,Volume 2, Issue 1,2016

Mallioris Panagiotis is a postgraduate student in the Industrial Engineering and Management Department of the International Hellenic University. Mr Mallioris has participated for three years into the autonomous vehicles student team and elaborated on data processing and management tools. He has also participated in Kleemann's young engineer program.

Sidiropoulos Athanasios is currently a PhD candidate in the Department of Mechanical Engineering, AUTH**.** He is an Electrical and Computer Engineer and his PhD research focuses on Digital Transformation of Supply Chains. His research interests include Supply Chain Management, Digital transformation, Process Optimization, Data Manipulation, Python and Web Programming.

Dr. Bechtsis Dimitriosserves as Assistant Professor at the Industrial Engineering and Management Department of the International Hellenic University, Greece. His scientific interests indicatively cover the fields of Information Technology, Database Management Systems, Internet Of Things, Industry 4.0, Autonomous Vehicles, Digital/Sustainable Supply Chains and Agrifood Systems, while he is focusing on the incorporation of intelligent autonomous vehicles and systems in the supply chain ecosystem.

Prof. Vlachos Dimitrios is the Director of the Laboratory of Statistics and Quantitative Analysis Methods, Logistics and Supply Chain Management (LASCM) of the Department of Mechanical Engineering of the Aristotle University of Thessaloniki (AUTh), Greece. He conducts research and consulting in the fields of supply chain management, logistics, applied operational research, combined transport systems management, business restructuring and strategy development.