USING CONVOLUTIONAL NEURAL NETWORKS FOR SOLVING PROBLEMS OF OBJECT DETECTION

Abstract. In the field of computer vision technology, which is related to the identification of objects in images and videos, the use of neural networks turns out to be an effective solution today. This work focuses on the implementation of artificial neural networks in the field of object detection, in particular, focusing on the integration of the YOLO algorithm into the architecture of convolutional neural networks. It was emphasized that in the case of this algorithm, the object detection task is formulated as a regression task, in which objects are identified within the image using a neural network analysis of pixel data. Each detected object is encapsulated by a bounding box that precisely outlines its position and dimensions. In addition, each recognized object is assigned several class probabilities that reflect the probability of belonging to different predefined categories. The main goal of YOLO is real-time object detection using a one-step detection process powered by a single neural network. This simplified approach allows rapid image analysis without the need for multiple passes, thus optimizing computational efficiency. The neural network was trained and tested on the 2011 and 2012 Pascal Visual Object Classes Project (PascalVOC) dataset. It was concluded that a further increase in the number of training iterations is not necessary given the level of recognition achieved, as a further increase in the number of training iterations will increase the reconfiguration of the neural network to the training data set and impair the transfer of the learned features to the test data set.

Keywords: image recognition, deep Learning, object localization, feature extraction, bounding boxes, class probabilities, real-time detection, regression analysis.

Introduction and statement of the problem. Recently, in the field of computer vision, artificial neural networks are gaining more and more popularity for solving the tasks of object detection. This technology has proven itself particularly
well in the field of autonomous driving, for example, recognizing road signs. However, the implementation of this concept is associated with a number of difficulties. In order to integrate the object detection technology into the neural network architecture, it is necessary to find a suitable method of object recognition in the image, as well as a suitable network architecture and acceptable hyperparameters. In addition, to train a suitable network, a collection of images must be manually created and annotated, as there is no labeled database of target images or videos. Also, a neural network usually needs a large volume to do this, so it is needed to find a way to keep this collection as small as possible. As a result, a special type of artificial neural network is required for image classification, which is an integral part of the object detection process. This type of neural network is Convolutional Neural Networks (CNN), which can analyze image features. Such neural networks differ from traditional neural networks mainly by special layers. These layers include the convolutional layer, the fully connected layer, and the union layer. The structure of convolutional neural networks consists of an input layer, followed by a combination of several convolutional and pooling layers, then one or more fully connected layers, and finally an output layer. This output layer is again responsible for categorizing the input image. Compared to the traditional neural network, the feature map can be seen with the output of the neuron and the kernel of the convolutional layer with weighted connection. These features provide the potential for effective classification and categorization of images and videos with the further goal of detecting objects on them.

Analysis of recent research and publications. Nowadays scientists have made a significant contribution to the development of methodologies for detecting objects in deformed convolutional networks. A number of studies were conducted to solve this problem.

In [1], a pairwise convolutional neural network-transformer (PCT) was suggested. The model fully utilizes the object detector and contains rich contextual information. Pairwise CNN features from the backbone CNN were obtained. These functions are combined with the pairwise transformer functions to improve pairwise representations. Advanced representations are superior to using CNN and transformer functions alone. In addition, global transformer functions provide valuable contextual clues. The experimental results show that the previously neglected CNN features still have a significant advantage. Compared to other methods, the model achieves competitive results on the HICO-DET and V-COCO datasets.

The study [2] focused on the optimization of the Faster R-CNN model for object detection using the KITTI dataset, with special emphasis on the detection of objects such as cars, pedestrians, and cyclists. Through complex training and model tuning, Faster R-CNN has been tuned to adapt to a variety of problems present in the dataset. The benchmarking also used YOLOv3 and YOLOv5 as benchmarks to
determine the relative strengths and weaknesses of Faster R-CNN. The results highlighted that although Faster R-CNN achieved outstanding accuracy, it lagged behind in speed, making YOLO models more suitable for real-time scenarios. The research was aimed at practical solutions regarding the selection and optimization of object detection models for real autonomous driving applications.

In addition, it is worth noting the works of the following scientists: Makinde Tobi [3], Sun Peize, Zhang Rufen, Jiang Yi, Kong Tao, Xu Chengfeng, Zhan Wei, Tomizuka Masayoshi, Yuan Zehuan, Luo Ping [4], Wang Jun, Kang Xining, Wang Jingjing, Zhang Xin, Ma Zhiyuan [5], Dingra Nikunj, Kumar Utkarsh, Mittal Abhishek, Yadav Sarita [6], Zhang Shenchuan, Yu Songlin, Ding Haixin, Hu Jie, Cao Lijuan [7], Pu Chengdao, Ju Liuxue, Gao Fan, Yu Jun [8], Anjali P., Kumar T., Lagisetti Ravi [9], Stoilkovic Dusan, Nejkovic Valentina, Yevtych Dusan [10], Butler Justin, Leung Henry [11], Xie Qiang, Zhou Daying, Tang Rui, Feng Hao [12], Hong Jixuan, He Xueqin, Deng Zhaoli, Yang Chenhui [13], Dong Wenbo [14], Yuhandri, Yanto Mousli, Nowri Eka [15] and others.

However, taking into account the above-mentioned scientific documentation, the issue related to object detection methodologies in deformed convolutional networks still remains under-researched and needs further elaboration.

**Setting objectives.** The purpose of the work is to study the principles of object detection using convolutional networks.

**Presentation of the main research material.** The goal of a system whose function is object recognition, which uses machine learning methods, can be "learning" based on the collected data, which is called "supervised learning". Important for this are data sets consisting of images on which the "solution" of the problem is already marked with the so-called Ground Truth Label. In the case of object recognition, this means that the objects in the images in the dataset are marked as bounding boxes. Supervised learning basically consists of approximating a function using examples. When applied to object recognition, this means that the information contained in example images is abstracted, and decision criteria for, for example, the presence, localization, and classification of objects in the images are inferred. If these criteria are successfully obtained, the resulting object recognition model should be able to process new unknown images and localize and classify the corresponding objects in them.

In the process of training a neural network, it is important to label data sets, for which in this case it is appropriate to use the method of unitary coding (One-Hot Encoding). Its principle consists in assigning to each category a vector of zeros and ones. The dimensionality of the vectors corresponds to the number of possible different categories. Each category is assigned a certain index in this vector. This means that the vector of this category consists entirely of zeros; since, according to this principle, the sum of the values of the vector is equal to one, the vector can also
be evaluated as a given, well-defined probability distribution of the corresponding category. If the category is added later, the advantage is that only one zero-valued dimension needs to be added to the existing vectors. Schematically, the essence of unitary coding is depicted in Figure 1.

<table>
<thead>
<tr>
<th>Class of the target object</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit code vector</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig.1 Graphic representation of the method of one vector of unitary coding. Each class of objects is assigned one vector, which differs in dimension, where 1 is located. Since the sum of the elements of the vector is 1, it can be viewed as a class assignment probability distribution.

In addition, this means that the normalized values of the original level can also be considered as a vector of probability distributions.

The cost function defines the error between the result of the neural network and the correct result. There are many different options for the type of error calculation. One of the most commonly used methods is to use the root mean square deviation (L2 distance), which is defined as:

\[
C = \frac{1}{2n} \sum || \xi - d(\xi) ||^2
\]  

where \( n \) is the amount of training data, \( \xi \) is the input data result vector, \( d(\xi) \) is the desired output for input data \( \xi \).

After determining the cost function, the goal of the learning process is to minimize the calculated error. So-called optimizers or optimization functions are used for this minimization process. A common method for this is the use of Gradient Descent algorithms. This method incrementally updates the weights and biases in the network to find the minimum error. The values are updated using backpropagation, which consists of applying a chain rule to the output of the cost function. A key point of this algorithm is the realization that the gradient of one layer can be expressed in terms of the gradient of the next layer. This means that if the
gradient of a layer is defined, the gradients in the entire network can be determined using recursive propagation.

The next step is data augmentation, which is used to augment a small training dataset with as little effort as possible and without the need for new data. The existing data is slightly modified to get the new data. For example, the image may be mirrored, rotated, or distorted. Also, only the cropped part of the source image can be used as a new file. If this is done for each file in the dataset, its size can be multiplied without requiring new data.

Detecting objects in an image is not the same process as classifying the image itself. The difference between object detection and classification is that object detection algorithms also analyze the position of the object in the image. This is often represented by a bounding box. In addition, object recognition should also be able to find multiple objects in an image. This cannot be solved directly with CNN because the number of objects in the image cannot be determined in advance and therefore the output vector is of variable length. An initial attempt to solve this problem was to identify different regions of interest in an image and apply a CNN to each of these regions for classification. The problem with this experiment is that the objects can be in different places in the image, they can have different aspect ratios, or they can be different sizes. Therefore, it was necessary to define a large set of regions to cover all possibilities. Techniques such as region-based convolutional neural networks (R-CNN) or You Look Only Once (YOLO) have been developed to solve this problem.

The purpose of the approximation is to ensure that the function can generate the correct output in the form of predictions from input data in the form of images. The outputs contain defined bounding boxes, classes, and confidences that perfectly match the ground truth labels. The function to be approximated contains a series of weights that are gradually adjusted during the learning process. The purpose of this adjustment is to produce an output for each input from the training data set that is as close as possible to the ground truth labels of the input data. Given a large amount of training data, by optimizing the weights, the model can abstract information from the input data and use this abstract data to make decisions about the presence and type of objects. Ideally, the decision framework is so abstract that it can also be applied to external inputs.

In order to compare different object recognition models and evaluate new techniques and approaches in practice, several metrics with different threshold values are often considered together. An example of this is the indicator $\text{mAP}^{\text{IoU}} = 0.50$, where mAP ((Mean) Average Precision) is calculated with an IoU threshold of 0.5. In addition, it is possible to include only objects in a certain size range in the calculation to analyze whether the object detection system provides better results in some size ranges than in others.

In this work, the main attention will be focused on the principles of object detection using convolutional neural networks based on YOLO algorithms. In the
case of these algorithms, object detection is considered as a regression problem in which objects are determined from the pixels of the input image using a neural network. The recognized object is described by the so-called bounding box, which clearly defines its position and size. In addition, as many class probabilities are determined for each recognized object as there are possible classes in the corresponding program. For each class, the probability value of the class is determined, the sum of which is equal to one. The detected object belongs to the class with the highest class probability. The main goal of YOLO is to detect objects in real time. Due to the properties of the one-stage object detector, only one neural network is used for the entire process of object recognition, which is performed for each image, therefore the image is viewed only once (You Only Look Once). This means that YOLO-based models can be thoroughly optimized according to their object detection performance.

In contrast, Regional Convolutional Neural Network (RCNN), Fast RCNN and Faster RCNN use the concept of two-stage object detectors. In the first step, regional proposals are created, that is, proposals for areas in which objects can be located, after which these areas are used, and the content is considered as a classification problem, that is, the class of the depicted object is determined.

When applying the YOLO algorithm, a neural network is used to make predictions about objects in the input image. To understand the processes in YOLO, the structure of the ground truth labels and predictions, i.e. the output of the neural network, must first be explained: The image to be processed is divided into a grid of cells $S \times S$. The cell that contains the center of the object in the image is responsible for object recognition, so ideally each object is defined by exactly one cell. $B$ bounding boxes belong to each of these cells. Number of cells $S$ and restrictive frameworks $B$ is chosen arbitrarily in the design of the YOLO object detection model.

The bounding box is described by coordinates $(x, y, w, h)$ and reliability assessment. Coordinates $(x, y)$ describe the center of the bounding box as an offset relative to the cell to which they belong. Coordinates $w, h$ describe the height and width of the frame relative to the entire image. Bounding boxes can be used to define the position and size of objects in images. The last value that fully describes the bounding box is the Confidence Score. It describes whether the corresponding bounding box contains an object and how well it matches that object. Formally, the assessment of the confidence of the bounding box is defined as follows:

$$
Confidence = Pr(Object) \times IoU_{truth_{pred}}
$$

$Pr(Object)$ — the objectivity indicator, which determines how likely it is that the frame contains an object. $IoU_{truth_{pred}}$ determines how accurately the dimensions and position of the box compare to the dimensions of the existing object.

In order to train an object detection model with YOLO, a ground truth must be created for each image in the training and evaluation dataset. This contains the appropriate definition of bounding boxes suitable for objects for each responsible
grid cell. The fundamental truth, thus, is determined by the marked coordinates of
the objects and the following optimal estimates of the reliability of the frames:

- no object in the frame: the confidence score is 0 because $Pr(\text{Object}) = 0$;
- the object is available: $Pr(\text{Object}) = 1 \rightarrow \text{Confidence Score} = 1 \times IoU^{\text{truth}}_{\text{pred}} = IoU^{\text{truth}}_{\text{pred}}$;
- the object exists exactly in the coordinates $x, y, w, h$ of the box: $Pr(\text{Object}) = 1; IoU^{\text{truth}}_{\text{pred}} = 1 \rightarrow \text{Confidence Score} = 1 \times 1 = 1$.

Another component of ground truth, and therefore prediction, are class
probabilities, the task of which is to describe to which class the recognized object
belongs, and which are determined not for the bounding box, but for the grid cell. In
YOLO, class probabilities are determined for each cell $C$ (number of possible
classes). They determine whether the cell contains an object of the corresponding
class. For fundamental truth, these values are the result $Pr(Class_i \mid Object)$.
The reason for defining by grid cell and not by bounding box is that each cell is
responsible for defining an object - so all bounding boxes of a cell must recognize
the same object.

A confidence score for a rectangle corresponding to a class can be calculated
based on the cell's class probability and the confidence score of the bounding box.
This is necessary for calculating the value of losses, according to which the model
is optimized. The reliability score for a particular class is determined as follows:

$$
\text{ClassCond.Prob.} = Pr(Class_i \mid Object) \times Pr(Object) \times IoU^{\text{truth}}_{\text{pred}} = Pr(Class_i) \times IoU^{\text{truth}}_{\text{pred}}
$$

With the structure described, the fundamental truth of the YOLO model
consists of values $S \times S \times (B \times 5 + C)$, which are modeled in the form of a tensor.

To create an object recognition system, a CNN is modeled that is capable of
generating a tensor in a form described as a prediction. The model is optimized
during the learning process, during which the weight of the model is gradually
adjusted. The goal of this process is to make the prediction for each input image as
close as possible to its ground truth label. When a model is optimized with many
training images, it "learns" to abstract the information and derive decision criteria
from it, which ideally can also be applied to other, non-training images. CNN
gradually reduces the resolution of the input image to size $S \times S$, simultaneously
increasing the channels, for example, from three color channels to $(B \times 5 + C)$.
Thus, the network generates an output tensor of the form $S \times S \times (B \times 5 + C)$.

Evaluation and comparison of various object detection methods is carried out
using unified data sets. They provide training and test datasets with necessary
annotations for supervised network training.

The Pascal Visual Object Classes Project (PascalVOC) provides standardized
image datasets and programming interfaces for the classification and detection of
objects in images. An overview of the size of the datasets for 2011 and 2012 can be
found in Table 1. These datasets are used to train and test methods that determine the presence or absence of 20 object classes in a classification task or the same 20 classes in an object detection task. Images in PascalVOC datasets are unstructured thematically and in terms of record type. Images of individual objects or groups of objects, as well as entire scenes, are available.

**Table 1**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Learning</th>
<th></th>
<th>Audit</th>
<th></th>
<th>Testing</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Image</td>
<td>Objects</td>
<td>Image</td>
<td>Objects</td>
<td>Image</td>
<td>Objects</td>
</tr>
<tr>
<td>PascalVOC 2011</td>
<td>5603</td>
<td>12605</td>
<td>5120</td>
<td>12632</td>
<td>9864</td>
<td>26143</td>
</tr>
<tr>
<td>PascalVOC 2012</td>
<td>14083</td>
<td>27308</td>
<td>11634</td>
<td>27673</td>
<td>6847</td>
<td>7893</td>
</tr>
</tbody>
</table>

In the first experiments, the development of the network will be investigated during training at different learning rates with an increasing number of iterations. The graphs in Figure 2 show the change in average accuracy as a function of the number of iterations for a high learning rate of 0.001 and a low learning rate of 0.0001. The CNN was trained on the PascalVOC 2012 dataset with a total of 150,000 iterations, with a snapshot of the neural network saved every 10,000 iterations and then tested.

Both curves have a characteristic shape in which the increase in recognition performance per iteration (as measured by mAP) decreases at higher iteration levels until a plateau is reached at which no further improvement occurs as the number of iterations increases. In the left graph, for a learning rate of 0.001, further training no longer improves the network's recognition performance after 40,000 iterations, in this experiment it fluctuates by only 65.9% within ±0.1%.

![Fig. 2 Comparison of mAP with the number of training iterations of training processes with different learning rates](image-url)
The right plot for the lower training level of 0.0001 shows a similar curve. A significantly worse recognition performance at 10,000 iterations is found (50.86% at $\eta = 0.0001$ vs. 59.61% at $\eta = 0.001$), which is a direct result of the lower learning rate. Due to the smaller steps of adjusting the network weights with a lower learning rate, the neural network needs more iterations to learn the relevant features. However, after approximately 40,000 training iterations, no further improvement in recognition performance was achieved. The best mean average accuracy here is only 62.6% ±0.1%, which is over 3% less than when trained with $\eta = 0.001$.

Conclusions. In this document, an overview of object detection methods in video and images using convolutional neural networks was made, in particular, a detection method based on a neural network using the YOLO algorithm.

When using CNN architecture models with the implemented YOLO algorithm, the task of object detection is formulated as a regression task, in which objects are identified inside the image using a neural network analysis of pixel data. Each detected object is encapsulated by a bounding box that precisely outlines its position and dimensions. In addition, several class probabilities are assigned to each recognized object, which reflects the probability of belonging to different predefined categories. CNN models can solve the identification of deformations, the creation of instructional/training programs. Compared to other CNN algorithms, YOLO has many advantages in practice. As a unified object detection model that is easy to build and train according to its simple loss function, YOLO can train the entire model in parallel. The YOLO algorithm also generalizes the object representation better compared to other object detection models and can be recommended for real-time object detection as a state-of-the-art object detection algorithm.

Prospects for further research are the development of a more effective methodology for detecting objects using deformed convolutional networks.

References:


