# THE USAGE OF ARTIFICIAL INTELLIGENCE IN RECOGNITION OF EMBOSSED NUMBERS ON BILLET

Petra SVOBODOVÁ, Antonín TOMEČEK, Jan KRÁTKÝ, Hana ŠPAČKOVÁ, Miroslav KLUS

*VSB - Technical University of Ostrava, Ostrava, Czech Republic, EU, petra.svobodova@vsb.cz*

https://doi.org/10.37904/metal.2021.4286

**Abstract**

The article is focused on the recognition and localization of numbers on steel billets. Serial numbers are embossed to each billet. Our automated solution allows product identification without human interaction. There are several problems caused by embossing to the hot steel. First, the numbers are not clearly visible. There is a lot of noise around the serial number which causes shadows and reflections. Next, the surface of the billet is rough with grooves and ridges. These issues affect object detection. As a part of the 4th Industrial Revolution, artificial intelligence and neural networks are used to automate production. Object recognition identifies which numbers are presented in the image. Another problem occurs when the serial number is located anywhere on the billet surface. The aim is to detect multiple objects in the scene using a single neural network. Our proposed solution is based on an extremely fast and accurate model from a class of deep learning algorithms. To localize and identify individual numbers, the You Only Look Once (YOLO) algorithm is implemented. It predicts bounding boxes and assigns classes from category 0 – 9. The approach is fully automatic and detects embossed numbers in real time. A custom dataset and annotations for train the model is created. Due to the lack of training images, data augmentation is used to extend a dataset by increasing the amount of data.

**Keywords:** Artificial intelligence, recognition of embossed numbers, object detection, YOLO

## 1. INTRODUCTION

Object recognition and detection are fundamental techniques of computer vision. They allow the computer to classify objects and estimate their position in the image. It is used for autonomous driving or bin picking, where the robot moves with the components using a 3D camera. The benefits of computer vision are encountered in our daily lives. In industry, factories have automated production based on image processing. The machine vision solution detects components and controls the shape, dimensions, and surface of products. In healthcare, it helps doctors identify tissues, organs and bones and reconstruct 3D model of the human body. In agriculture, the quality of food is controlled. Computer vision is also focused on number detection and recognition. For example, identification vehicles according to their license plates. This technology is already implemented in traffic and safety applications. This task is challenging due to changes in weather, daily changes, the speed of cars and the different shape of the plate. The second task is to recognize house numbers. Google Street View creates a map by recognizing multi-digit numbers in photos taken with a 360° panoramic camera. Handwritten digits from documents such as phone numbers or important data can also be read and identified. In the industry, each component has a serial number, and each product has an expiration date. All of the above have several things in common. First, the entire numeric sequence is localized in the image. Furthermore, the individual numbers are separated and recognized. Approaches and solutions to solve these tasks are described in the following chapter.

This article presents an approach to the detection and recognition of embossed numbers. Our solution is real time and fully automatic. The previous requirements are necessary in many computer vision applications. In the industry, each product has a unique identification number. For example, serial number, expiration date on

food and medicine, digits on tires and electronic components. Usually, these sequences of numbers are controlled by human workers. This process is time consuming and usually leads to inattention mistakes. The problem is fast and continuous production in many industrial factories. The production lines are in operation 24/7. Workers cannot concentrate on a huge number of digits. The solution is machine vision. The camera is usually placed above the production line and takes pictures of the product. In our case, the camera captures a steel billet. The software then reads and identifies the embossed numbers. The whole process is automated. Its advantage is lower error rate and faster production.

## 2. RELATED WORKS

Earlier, discriminative classifiers were used to recognize digits. In [1], Support Vector Machine recognizes handwritten numbers. Although the results are satisfactory, these methods are not resistant to illumination changes. Therefore, hybrid methods based on the CNN-SVM model were then developed [2]. They use CNN to extract features and SVM as a classifier. These days, convolutional neural networks represent state-of-the-art approach to number recognition [3]. These CNNs are trained on hundreds or thousands of samples and achieve impressive results. Typically, only one unknown number is presented in the image. This image is sent as input to CNN. The output is a class prediction.

Objects can be detected using the above methods only if the numbers still have the same position in the image and each sample contains only one digit. The problem occurs when the serial numbers are located anywhere on the surface of billet. Recognition is not enough at this point. The number must first be localized by using object detection methods. Binary or morphological operations can be used to separate individual numbers. However, this approach is not appropriate in our case. The surface of the billet is rough and contains shadows and reflections that influence the detection result. The main advantages of neural networks are noise resistance and illumination resistance. Neural networks have many types of architectures for object detection. The most famous detectors are Fast R-CNN (Fast Region-Based Convolutional Network), SSD (Single Shot Detector) and YOLO (You Only Look Once). Compared to SSD, YOLO looks to the image only once and computes a feature map. Also, YOLO is faster and more robust with higher accuracy. [4] Based on this survey, it was decided to implement YOLO as our object detector to localize and classify embossed numbers.

## 3. YOU ONLY LOOK ONCE

You Only Look Once, a real-time object detector, was introduced in 2016 by Joseph Redmon [5]. He came up with a completely new approach. YOLO is a single neural network that predicts bounding boxes to estimate location and class probabilities. However, the algorithm is extremely fast, it is also very accurate. Only one pass forward through the network is applied to the entire image. YOLO has several modified architectures. Our solution is based on YOLOv3. [6]

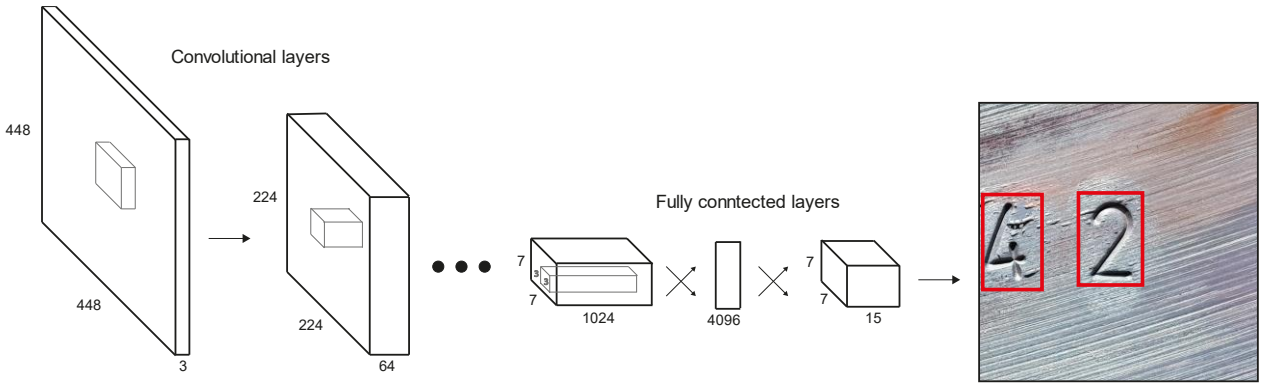


**Figure 1** YOLOv3 architecture

The YOLO implementation consists of the following steps. First, the YOLO architecture is based on convolutional and classification parts. The model used to detect embossed numbers is shown in **Figure 1**. There convolutional layers are shown, followed by pooling layers. To reduce the dimensions of filtered images after convolution, the maximum pooling method is used. At the end of the convolution part is an operation called flattening. The result of the flattening is a one-dimensional vector. Each pixel of the image is converted to one neuron. These neurons form the input layer which servers as input to the classification part of neural networks. There is a dense or fully connected layer which connects neurons from one layer to another. In our case, the output layer has ten neurons. Each neuron represents an embossed class of numbers 0 – 9.

YOLO divides the input image into a grid $S \times S$. Each cell predicts a $B$ number of bounding boxes. In our case, the parameters are set to $S = 13$ and $B = 5$. Each cell is responsible for objects whose centers belong to a particular grid cell. Each bounding box is estimated by a vector:

$$y = [p_c, b_x, b_y, b_h, b_w, c_n]^T$$

The $p_c$ is an object parameter which takes only two values. If there is an object presented $p_c = 1$. Otherwise, $p_c = 0$. The parameters $b_x, b_y$ represent the center of the bounding box. The $b_h, b_w$ represent the height and width of the bounding box, respectively. The $c_n$ parameter tells whether there is an object of a particular class in the bounding box. The parameter $n$ represents the number of classes. In our case, $n = 10$. Then $c_1$ is for class one, $c_2$ is for class two, $c_3$ is for class three etc. If the bounding box captures a number from classes 0 – 9, the particular $c_n$ is set to one. Otherwise, $c_n$ is set to zero. An example of an input image and vectors is shown in **Figure 2**. As can be seen, the grid divides an image into cells. The cells marked in orange and green contain and object. The rest of the cells are empty. Such a cell is marked in purple color. Then each vector describes the relevant cell. The connection of all vectors is represented in the tensor. [7,8]

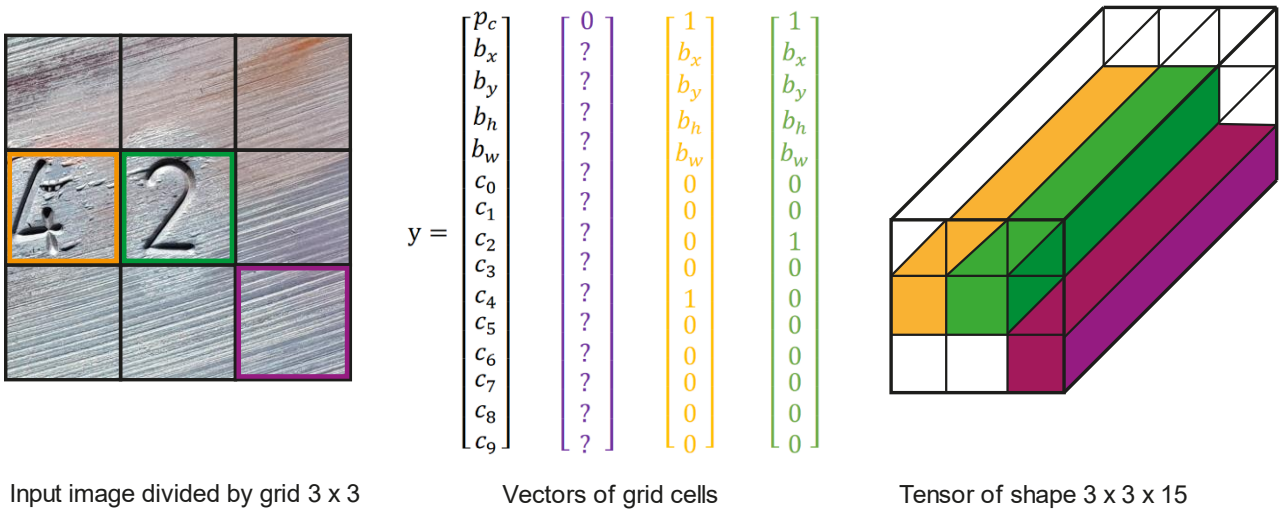


**Figure 2** On the left is the input image divided by a 3x3 grid. Cells that are marked in orange and green capture the object. On the other hand, the cell marked in purple does not contain any object. In the middle of the image, vectors are shown that describe the objects in the cells and their bounding boxes. These vectors form the output tensor of the YOLO architecture which is shown on the right.

Typically, there are multiple overlapping objects in the scene. But each grid cell is responsible for one object. The problem occurs when a cell contains more than one center points of two different objects. This means that the cell should be responsible for at least two objects. The technique that solves this problem is called anchor boxes. Anchor boxes allow a single grid cell to store information about multiple objects. To simplify the solution, the grid cell vector that defines multiple objects will be longer.

Anchor boxes have a defined width and height. This is because some objects are taller, such as pedestrians. Otherwise, some objects are wider, such as cars. YOLO tries to find objects that match these parameters. For example, if dimensions are defined for nine anchor boxes, each grid cell will predict nine anchor boxes. In the above case, when the image is divided by a grid $3 \times 3$, the total number of predictions is $3 \times 3 \times 9$. The total number of anchor boxes is 81 per image. Typically, most of the predicted anchor boxes has a low probability value. Non-maximum suppression is used to get rid of these unnecessary predictions. In the first step, all predictions with probability values lower than the specified threshold are removed. In the second step, the Intersection over Union (IOU) is used to find the best fit detections. [9,10]

## 4. EXPERIMENTAL PART

The numbers are embossed on a steel billet. It causes problems with visibility and subsequent reading. Parts of the embossed numbers are missing or overlapping. As can be seen in **Figure 3**, the background is different for each image. There are grooves and ridges. The result is influenced by noise and shadows.

### 4.1. Dataset

Embossed numbers are different compared to handwritten digits. That is why a custom dataset was created. To train the network, images representing the numbers 0 – 9 were used. To create the dataset, the individual embossed numbers were manually labeled. This was made from real images. These images were captured with an industrial camera. The camera is placed above the production line. Due to the lack of training images, data augmentation is used. This technique extends the dataset with image transformation operations such as rotation, width and height shifting, changing the zoom, intensity and flipping. CNN can learn from several examples. Image data generation is used only for training dataset, not for validation or testing.

**Figure 3** Example of dataset of embossed numbers

### 4.2. Training part

In our case, the YOLO architecture can be described as follows. The input images have a size of $416 \times 416$ and the number of channels is 3. It works with RGB images. The parameters are set to $S = 13$, $B = 9$ and $n = 10$. Then the vector $y = 15$. The pipeline below (1) shows the process of the YOLO algorithm. The parameter $m$ represents the number of samples. The samples pass through a deep neural network. The output is a specific number of bounding boxes for each cell. The number of predictions after the first pass is $13 \times 13 \times 9 = 1521$ boxes. The boxes are reduced to keep only a few interesting of them.

$$IMAGE(m, 416{,}416{,}3) \rightarrow YOLO\ CNN \rightarrow ENCODING(m, 13{,}13{,}9{,}15) \quad (1)$$

Technologies such as GPU, CUDA, cuDNN (Cuda Deep Neural Network Library), OpenCV and OpenMP are enabled before the start of the training. Embossed number recognition was trained on graphical card NVIDIA GeForce MTX250 with CUDA version 11.2. The initial weight file called $darknet53.conv.74$ was used to train custom data. The output file $yolov3.weights$ is then used for detection. Next, the parameters for training the network were set in configuration file $yolov3.cfg$ as follows. The dimensions of the images are $416, 416$. The number of classes is set to $10$. The $anchors$ take the following values - $19, 37,\ 25, 41,\ 33, 55,\ 37, 56,\ 42, 64,$

$40,70,\ 46,74,\ 54,85,\ 86,135$. The number of filters is computed according to the formula $filters = (classes + 5) * 3$. The total training time of the network with the above parameters for $2000$ iterations was approximately 12 hours.

## 5. RESULTS

The resulting YOLO detection is shown in **Figure 4**. Each embossed number is marked by a bounding box. The colors of the bounding boxes represent different classes. In addition to the bounding box, the result includes the class name and the probability value.

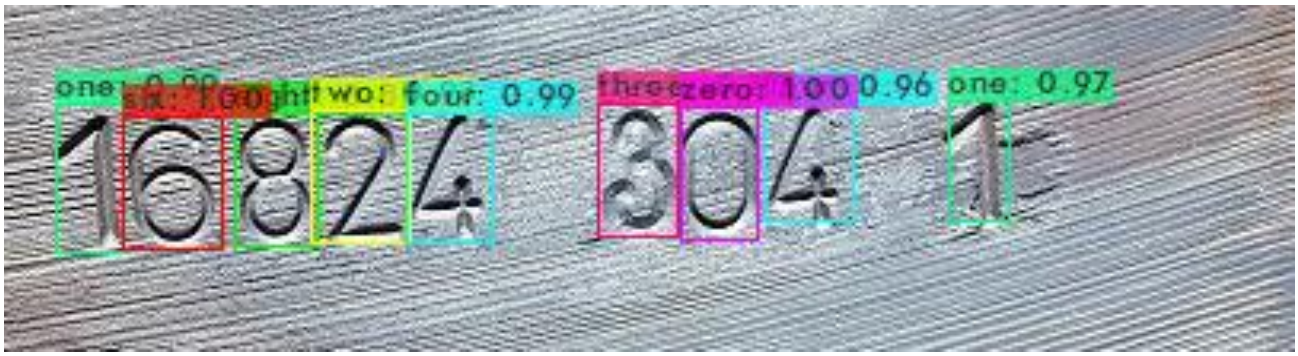


**Figure 4** Detection of embossed numbers

The classification identifies the object. On the other hand, localization predicts the coordinates of the bounding box around the detected object. When evaluating our model, the coordinates of the predicted bounding box must be compared with ground truth. A measure technique called mean Average Precision (mAP) is used to evaluate object detection algorithms. In other words, mAP is the average of precisions where precisions represent how accurate our prediction is. Equation is presented in (2). It compares the intersection of our prediction with the ground truth. This technique is called Intersection over Union (IoU). If the intersection is greater than the threshold, the prediction is True Positive (TP). Otherwise, it is False Positive (FP) detection.

$$precision = TP/(TP + FP) \tag{2}$$

As can be seen in **Figure 5**. The red line represents the metric mAP used to measure the performance of our model. The mAP function achieves $100\%$ at the end of training. The blue line represents the loss function. You can see a loss graph during training. Finally, the loss of the YOLOv3 model is about $0.1880$.

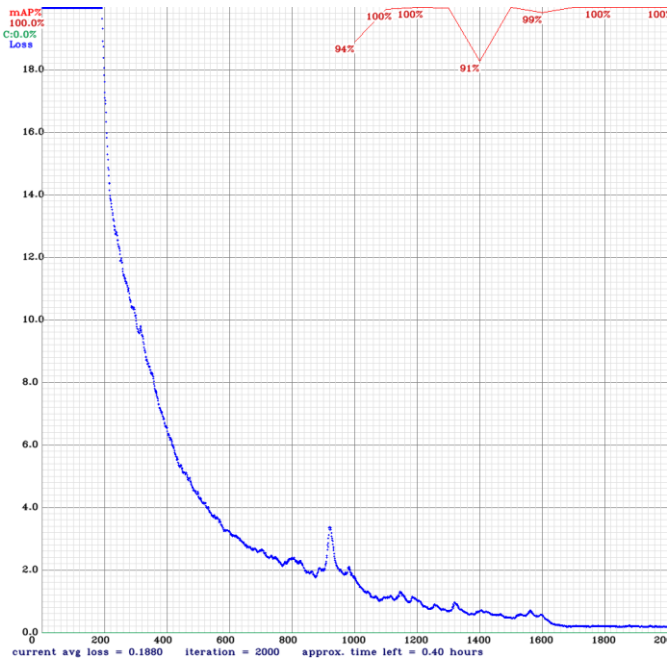


**Figure 4** Chart of YOLO prediction

## 6. CONCLUSION

In this paper, an approach to the detection and recognition of embossed numbers in steel billets was proposed. This method detects the number in real time and is fully automatic. The result is the recognition and pose estimation of numbers. Recognizing embossed numbers was quite a challenging task. The scene is influenced by ridges, grooves and shadows which is caused by embossing into the steel material. The numbers are sometimes incomplete and difficult to see even with the human eye. The solution is based on the state-of-the-art method of deep learning called YOLO (You Only Look Once). The advantage of YOLO is accuracy and speed. In contrast with other convolutional neural networks, YOLO sees each image only once. The YOLOv3 architecture was also described in this article. Then the experimental part summarizes the parameters of YOLOv3. Appropriate setting of the parameters is crucial for high detection accuracy. The custom dataset was created for training, testing and validation of the model. The mean Average Precision (mAP) achieves almost 100% after a thousand iterations. The trained model can successfully detect and recognize embossed numbers and will be used as part of a machine vision solution.

## ACKNOWLEDGEMENTS

***The work was supported by the specific university research of Ministry of Education, Youth and Sports of the Czech Republic No. RPP2021/50 and SP2021/71 and SP2021/23.***

## REFERENCES

[1] WU, M., ZHANG., Z. Handwritten Digit Classification using the MNIST Data Set. In: *Course project CSE802: Pattern Classification & Analysis*. 2010.

[2] NIU, X., SUEN, CH. Y. A novel hybrid CNN-SVM classifier for recognizing handwritten digits. In*: Pattern recognition*. [online]. 2012, vol. 45, no. 4, pp. 1318-1325. Available from: https://doi.org/10.1016/j.patcog.2011.09.021.

[3] PRATT, S., OCHOA, A., YADAV, M., SHETA, A., ELDEFRAWY, M. Handwritten Digits Recognition Using Convolution Neural Networks. In: *Journal of Computing Sciences in Colleges*. [online]. 2019, vol. 314, no. 5, pp. 40-46. Available from: https://dl.acm.org/doi/pdf/10.5555/3344038.3344042.

[4] KIM, J. -a., SUNG, J. -Y., PARK, S. -h. Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. In: *IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia).* [online]. 2020, pp. 1-4, Available from: https://doi.org/10.1109/ICCE-Asia49877.2020.9277040.

[5] REDMON, J., DIVALLA, S., GIRSHICK, R., FARHADI, A. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*. [online]. 2016. Available from: https://arxiv.org/abs/1506.02640.

[6] REDMON, J., FARHADI, A. YOLOv3: An Incremental Improvement. [online]. 2018. Available from: https://arxiv.org/abs/1804.02767.

[7] LAN, W., DANG, J., WANG, Y., WANG, S. Pedestrian Detection Based on YOLO Network Model. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA).* [online]. Changchun, China, 2018, pp. 1547-1551. Available from: https://doi.org/10.1109/ICMA.2018.8484698.

[8] MASUREKAR, O., JADHAV, O., KULKARNI, P., PATIL, S. Real Time Object Detection Using YOLOv3. *International Research Journal of Engineering and Technology (IRJET).* 2020, vol. 07, no. 03, pp. 3764-3768.

[9] DAVID, J., SVEC, P., FRISCHER, R. Support for maintenance and technology control on slab device of continuous casting. In: *Metal 2013: 22nd international conference on metallurgy and materials*. Brno: TANGER Ltd, 2013, pp. 1650-1655.

[10] FRISCHER, R., DAVID, J., SVEC, P., KREJCAR, O. Usage of analytical diagnostics when evaluating functional surface material defects. *Metalurgija.* 2015, vol. 54, no. 4, pp. 667-670.