Mineral Recognition Using YOLOv5 Technology /Ásványok Felismerés YOLOv5 Technológia Segítségével

Márton Gergely Herkules

***Abstract* — One of the most interesting types of neural networks are convolutional neural networks, one of the main applications of which is image processing. There are several types of image processing itself, but this paper will only cover object detection and classification. My task was the detection and identification of certain minerals and, above all, the reading of their weight from the digital display of a given immersive device via images. The gemstones themselves and the numbers on the digital display were detected by object detection. The rocks detected in the images were then processed as separate images by a classification model, which can give a more reliable value with little training data than the object detection method. The resulting system contains three separate neural networks, which can eventually be used to confidently identify individual gemstones.**

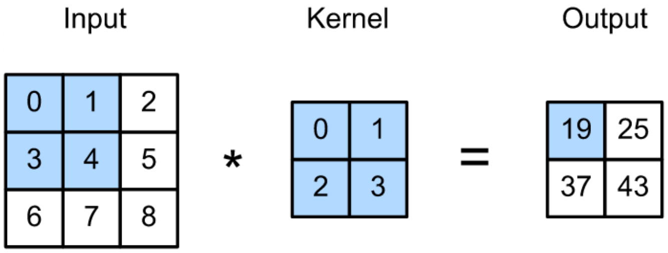
***Kivonat* — A neurális hálók egyik igen érdekes fajtája a konvolúciós neurális hálók, amiknek egyik fő felhasználási területe a képfeldolgozás. Magának a képfeldolgozásnak több fajtása van, de ez a dokumentum csak objektum detektációra és klasszifikációra fog kitérni. A feladatom egyes ásványok felismerése és meghatározása volt és emelett azoknak a súlyának leolvasása egy adott merő eszköz digitális kijelzőjéről képeken keresztül. Maguknak a drágaköveknek és a digitális kijelző számainak felismerése objektum detektációval történt. A képen érzékelt kőzetet pedig utána külön képként még egy klasszifikációs modell is feldolgozta, mivel az biztosabb értéket tud adni kevés tanítási adat esetén, mint az objektum detektációs módszer. Az elkészült rendszer három külön neurális hálót tartalmaz, amik segítségével végül magabiztosan meg tudja határozni az egyes drágaköveket.**

I. INTRODUCTION

THIS DOCUMENT PRESENTS MY SEMESTER HOMEWORK ASSIGNMENT WITHIN THE BME COURSE "DEEP LEARNING IN PRACTICE BASED ON PYTHON AND LUA"

II. CONVULUTIONAL NEURAL NETWORKS

Convolutional neural networks (CNNs) [1][2] are deep learning models designed primarily for image and sound processing tasks. Convolutional networks are highly efficient in that they are explicitly suited for spatial hierarchies and feature extraction. The basis of these meshes is the convolution operation, which is a mathematical operation that derives a third function from the sum of two functions. In image processing, this means applying a small window (usually a "filter" or "kernel") to a portion of the input image, i.e. multiplying the values of the image by the filter weights and then summing these values. This operation is repeated over the whole image, step by step, until the whole image is covered as shown in figure 1.



1. Figure: kernel usage

Most of these neural networks use many convolutional layers in sequence, so they can extract increasingly complex features from the image that may be useful for the task, such as edges, shapes, colors, or higher-level features. All these feature extractions make it possible for the system to recognize specific objects in some images. This is the main building block of classification and object detection itself.

III. METRICS FOR EVALUATING MODELS

True Positive (TP)

These are instances where the object detection model correctly identifies and localizes objects.

False Positive (FP)

These are cases where the model incorrectly identifies an object that does not exist in the ground truth or where the predicted bounding box has an IOU score below the defined threshold.

False Negative (FN)

FN represents instances where the model fails to detect an object that is present in the ground truth. In other words, the model misses these objects.

True Negative (TN)

Not applicable in object detection. It represents correctly

2023.12.06

rejecting the absence of objects, but in object detection, the goal is to detect objects rather than the absence of objects.

Precision

Precision [3][4] is a critical metric in model evaluation as it serves to quantify the accuracy of the positive predictions made by the model. It specifically assesses how well the model distinguishes true objects from false positives. In essence, precision provides insight into the model's ability to make positive predictions that are indeed accurate. A high precision score indicates that the model is skilled at avoiding false positives and provides reliable positive predictions.

$$precision = \frac{TP}{TP + FP}$$

Recall

Recall[3][4], also known as sensitivity or true positive rate, is another essential metric used in evaluating model performance, especially in object detection tasks. Recall measures the model's capability to capture all relevant objects in the image. In essence, recall assesses the model's completeness in identifying objects of interest. A high recall score indicates that the model effectively identifies most of the relevant objects in the data.

$$recall = \frac{TP}{TP + FN}$$

F1-Score

Is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance, considering both false positives and false negatives. This metric is particularly useful when there is an imbalance between positive and negative classes in the dataset.

$$\mathrm{F1} = \frac{2 * (precision * recall)}{(precision + recall)}$$

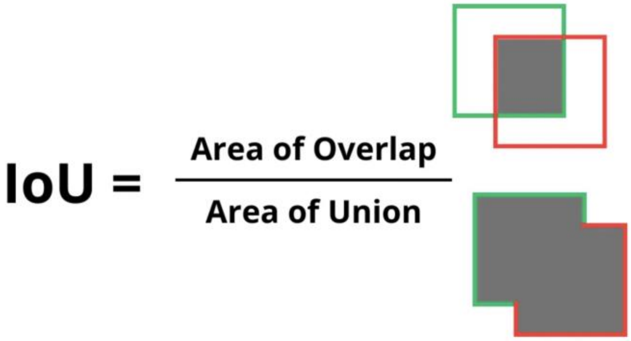
Intersection over Union (IoU)

Intersection over Union (IoU)[5], also known as Jaccard's Index, serves as a pivotal metric in the realm of computer vision, particularly for tasks like object detection and segmentation. It plays a crucial role in assessing the quality and accuracy of these models. The IoU is calculated by taking the intersection area of two bounding boxes and dividing it by the union area of these boxes as shown in figure 2. In mathematical terms, the IoU formula is expressed as follows:

$$\mathrm{IoU} = \frac{(Area\ of\ Intersection)}{(Area\ of\ Union)} = \frac{|A \cap B|}{(A \cup B)}$$

This formula essentially quantifies the degree of overlap between the predicted bounding box and the ground truth bounding box. It provides a numerical value that indicates how well the model's prediction aligns with the actual object location. A higher IoU score indicates a better match between the predicted and ground truth bounding boxes, signifying superior localization accuracy.

IoU is an essential metric because it measures the model's ability to precisely localize objects within images. When it comes to object detection, IoU evaluates how well the model identifies the objects' positions. In segmentation tasks, it assesses the model's capacity to distinguish objects from their backgrounds.



2. Figure: IoU

IV. System design

The system consists of three neural networks. Two YOLOv5m [6][7] object detection neural networks [8] with the same structure and one YOLOv5m-cls classification neural network.

In operation, a given image is processed by the system in such a way that first the number-detecting neural network searches for the numbers on the digital display of the scale in the given image. These are processed by the system based on their position and tell the exact number shown on the scale. In addition, the other object detection neural network looks for the gem in the image and identifies which class it belongs to and also "cuts" the detected object out of the image and passes it to the calibration model, which decides separately what it thinks is in the image. Finally, the output is the recognized number, which is the weight of the stone and the opinion and confidence of the two separate models as to which gemstone is in the image.

V. Implementation

Obtaining and preparing data

The data was all obtained from Roboflow or manually labelled by me. I had to prepare images for teaching in different ways, but for each teaching data I needed data augmentation[9][10]. For numbers I used all the usual augmentations as they can come up in any form during practical use of the system. These augmentation parameters are defined in a .yaml file [11]. For the gems I used my own self-defined augmentation steps, defined using Roboflow:

- Flip: Horizontal,
- 90° Rotate: Clockwise, Counter-Clockwise
- Rotation: Between -4° and +4°
- Shear: ±5° Horizontal, ±5° Vertical

This was important to avoid distorting the shape of the stones and changing their colour during the neural network training, as it is the only way to distinguish the types of stones in images. For object detection, the images were 640x640 pixels and for classification they were 320x320 pixels, as the convolutional neural network processing them is much smaller. The databases I have compiled and used can be found here: [12][13]

Training

Training the models is very simple. You need to specify the size of the images to be input to the model, the number of epochs to be taught and the data.yaml file, which contains the path to the database and the names and code of the classes in it. In addition, I specify that it should use a pre-trained YOLOv5m model in transfer learning and that it should have a patience of 5, which means that after each epoch it runs an evaluation of the model on the validation dataset to check if the model is evolving during the training, but if after 5 epochs it stops evolving, the model stops learning.

Evaluation

The model was evaluated on the test data set. The images extracted from the dataset are received by the model and the objects should be recognized on it. The test measures not only whether the neural network recognizes the objects, but how accurately it does so. Whether it hits the center of the objects and if not, how far the predicted object slips in space from the real object in the image. At the end of testing, the script returns precision, recall, confusion matrix, etc. broken down by classes.

Training and evaluation script

The download of the prepared databases, the teaching and evaluation of the models are available in the jupyter notebooks [14]. Running these will produce a file containing the weights of the trained model and its evaluations.

Use of the trained models

The use of trained models is implemented by Detection.ipynb. At the very beginning, it is necessary to specify the access to a specific image for the script (imagePath). After that, the script downloads the necessary class libraries and the weights of the previously trained neural networks. Derive the three necessary neural networks based on the downloaded weights. GetFrame is used to read the image so that it can be processed by convolutional neural networks. After all the conditions are given, getDetectedNumber processes the image and returns the found number, while getDetectedGemFrameAndPred returns the prediction of the found gem and its clipped location from the image. The cut part is then processed by the classifier and returns its prediction. Finally, the program displays the detected number, followed by the class and probability of the gem detected by the two models.

VI. Future plan

The designed system for recognizing precious stones and reading their weight was created in such a way that it gives the appropriate predictions. Future plans include that, based on the recognized weight and value, the system will provide the predicted value based on the current market. The market value of gems changes very often, and their value is not linear based on their weight, so this is not a simple question and will probably require continuous model updates.

References and Footnotes

[1] Tianmei Guo; Jiwen Dong; Henjian Li; Yunxing Gao "Simple convolutional neural network on image classification" https://ieeexplore.ieee.org/abstract/document/8078730

[2] Saad Albawi; Tareq Abed Mohammed; Saad Al-Zawi "Understanding of a convolutional neural network." https://ieeexplore.ieee.org/abstract/document/8308186

[3] Kemal Oksuz, Baris Can Cam, Emre Akbas, Sinan Kalkan; "Localization Recall Precision (LRP): A New Performance Metric for Object Detection" Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 504-519 https://openaccess.thecvf.com/content_ECCV_2018/html/Kemal_Oksuz_Localization_Recall_Precision_ECCV_2018_paper.html

[4] Yang Liu, Peng Sun, Nickolas Wergeles, Yi Shang "A survey and performance evaluation of deep learning methods for small object detection" https://www.sciencedirect.com/science/article/pii/S0957417421000439?casa_token=NnfhubznxpEAAAAA:_1ENG6fXPnmQKRJ7uAzzE__rgnLwyREH8BRdC-agYA4DNXthMZvj35CBLae-WB2x3JcExDA

[5] Shengkai Wu, Jinrong Yang, Xinggang Wang, Xiaoping Li "IoU-Balanced loss functions for single-stage object detection" https://www.sciencedirect.com/science/article/pii/S0167865522000289?casa_token=m_grltUfr68AAAAA:F1mYw9KDsyNNE9l9yi4O9QoHZKBfiAU5sZhWVAUgZGOLgW4QgAfb8WSVVlFhbEglqFM7dzg

[6] Juan Terven, Diana Cordova-Esparza "A Comprehensive Review of YOLO: From YOLOv1 and Beyond" https://arxiv.org/abs/2304.00501

[7] Xingkui Zhu, Shuchang Lyu, Xu Wang, Qi Zhao; "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios" Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, 2021, pp. 2778-2788 https://openaccess.thecvf.com/content/ICCV2021W/VisDrone/html/Zhu_TPH-YOLOv5_Improved_YOLOv5_Based_on_Transformer_Prediction_Head_for_Object_ICCVW_2021_paper.html

[8] Anamika Dhillon & Gyanendra K. Verma "Convolutional neural network: a review of models, methodologies and applications to object detection"https://link.springer.com/article/10.1007/s13748-019-00203-0

[9] Luis Perez, Jason Wang "The Effectiveness of Data Augmentation in Image Classification using Deep Learning." https://arxiv.org/abs/1712.04621

[10]Jason Wang, Luis Perez "The Effectiveness of Data Augmentation in Image Classification using Deep Learning." http://vision.stanford.edu/teaching/cs231n/reports/2017/pdfs/300.pdf

Other sources

[11]YOLOv5 Github repository: https://github.com/ultralytics/yolov5/blob/b94b59e199047aa8bf2cdd4401ae9f5f42b929e6/data/hyps/hyp.scratch-low.yaml

[12]Roboflow number database: https://universe.roboflow.com/legoproject/digital-numbers-vqqx2/model/3

[13]Roboflow Gem databases: https://universe.roboflow.com/msc-onlab-1/gem-9oe6g/model/1 https://universe.roboflow.com/msc-onlab-1/gem-q84tk/dataset/3

[14]Github repository: https://github.com/HerQTheToxic/DeepL