Real-time Train Wagon Counting and Number Recognition Algorithm.

Andrey Vavilin
Abbyy
andrey.korea@gmail.com

Andrey Lomov
EasyData
office@easydata.nl

Titkov Roman
Novosibirsk State University
r.titkov@g.nsu.ru

In this work we present an efficient solution for counting train wagons and recognizing their numbers using deep learning computer vision models. The proposed method is a good alternative for radio-frequency identification (RFID) method in terms of low cost and ease of use. Our system shows 99% accuracy in real-world scenarios, including corrupted wagon numbers and night shooting conditions. At the same time, the proposed method is capable to process video-stream in real-time speed without GPU-acceleration.

Keywords — object detection, number recognition, wagon counting, deep learning, computer vision

I. INTRODUCTION

No doubt railway system plays an essential role in transportation and logistics worldwide. Efficient managing of railway traffic requires every wagon to be tracked in order to improve the logistic speed and quality. There are more and more systems that allow automatic counting and identification of wagons as the train passes [1-5]. Nowadays there are two main ways to track wagons at railway stations: manual and radio-frequency identification (RFID) [3] methods.

A manual method usually requires one or more humans, who observe passing trains and write down wagon numbers. The accuracy of this method can be not really high, because human can be tired, get distracted or confused. There are also labor costs for these people.

The RFID method has better accuracy, but it has a lot of preparation costs. Every wagon must have an RFID marker and the maintenance costs are also high.

Currently, new ways of recognising numbers are emerging - the use of computer vision and deep learning [2,3]. We are going to use video cameras as an alternative to the approaches described above. This method has a low cost compared to other methods and achieves high accuracy [3,5]. The image-based approach is possible because each wagon has a painted number on the sides, and the correctness of the number can be checked using a checksum [6, 9].

In the wagon number recognition the main problem is variety on conditions, such as weather or light conditions. Also wagon numbers can be dirty or partially erased (Fig. 1) [2]. Sometimes there are additional digits next to the wagon number, so it may be a challenge to determine which digits are a part of the number.

Also the solution must be computation effective [7] in order to work on CPU at a good speed, and there are some tricks which will be described in this paper.

In our experimental evaluation, using 720x405 videos and images our approach achieved good results, i.e. 97.15% accuracy of the total number of recognised images and 99.4% accuracy of car number recognition when using multiple wagon images.

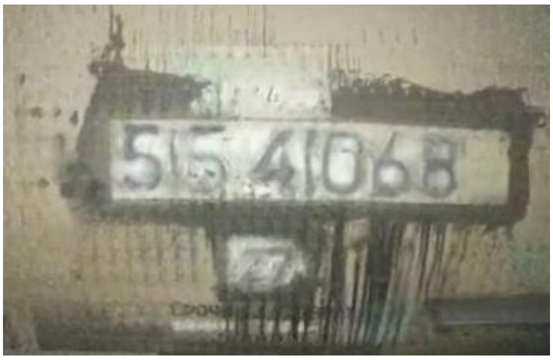


Fig 1: An example of a number with distortions

The remainder of this paper is organised as follows. The proposed methodology is presented in Section 2. The dataset is described in Section 3. The results of our experiment are presented in Section 4. The conclusion and future work are described in section 5.

II. PROPOSED APPROACH

The pipeline of the proposed method is shown in Fig.2. As an input our system takes a video stream which is processed as a sequence of independent frames. Every frame is passed to the classification step, which decides if there is a wagon. Frames with a wagon are passed to the number detection and recognition steps.

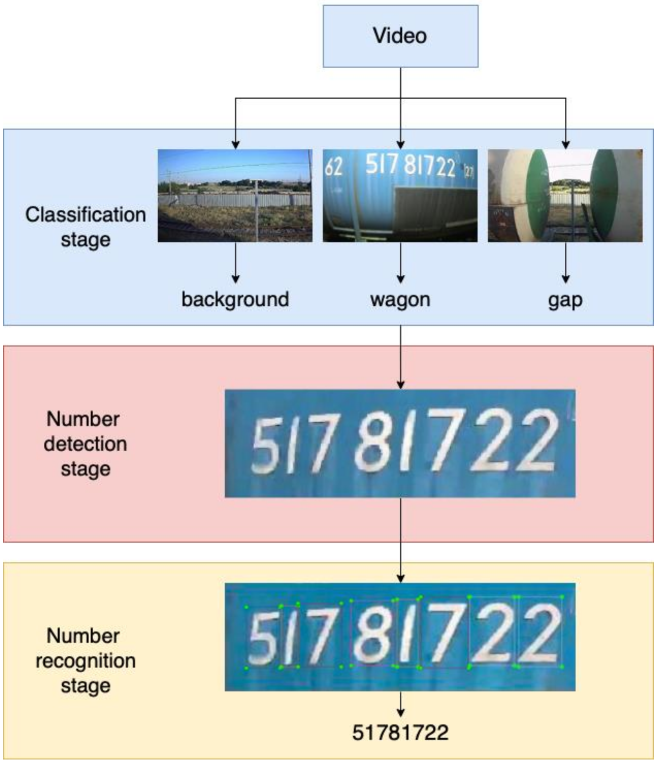
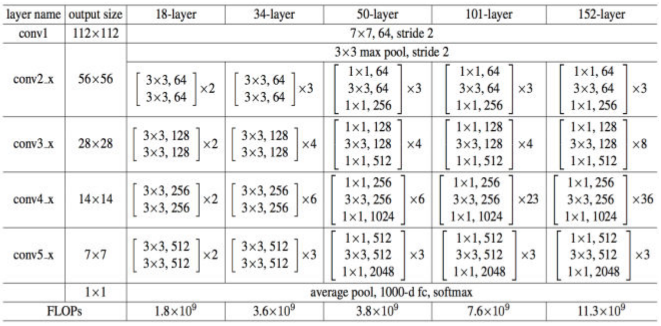


Fig 2: Pipeline of the proposed approach



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig 3: ResNet Architectures

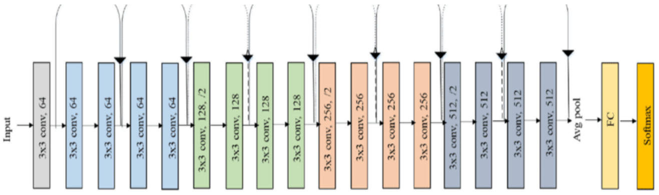


Fig 4: ResNet-18 architecture

A. Scene Classifier

First, every frame is processed by a deep learning model - scene classifier. We compared different architectures on the validation dataset and get the results presented in Table I. From ResNet-18 (Fig. 4), ResNet-34 and ResNet-50 architectures (Fig. 3) we chose the first one because it is a small and fast model, that achieves a similar score, that bigger models on the scene classification task. The ResNet-18 model can process up to 36 fps, while the bigger models work much slower.

Model	F1 Score	FPS
ResNet-18	99.2%	36
ResNet-34	99.4%	22
ResNet-50	99.7%	10

Table 1: Comparison Of Architectures

This model predicts what is in the image: a gap between wagons, a wagon or a background. Frames with gaps are used for wagon counting and wagon frames are passed to the recognition step. This step allows reducing the number of frames passed into the recognizer which significantly reduces the processing time. Frames without gaps and wagons are ignored.

Iterating throughout the frames gives us a number of wagons passed. Furthermore, considering an average number of frames per wagon it is easy to detect errors in counting and fix them.

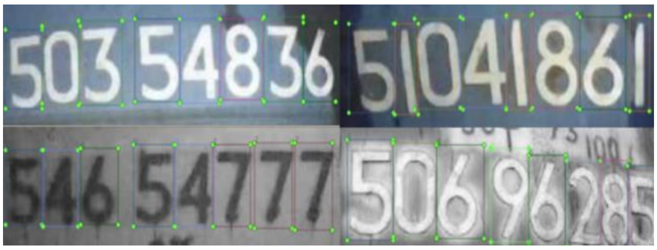


Fig 5: Digit Borders

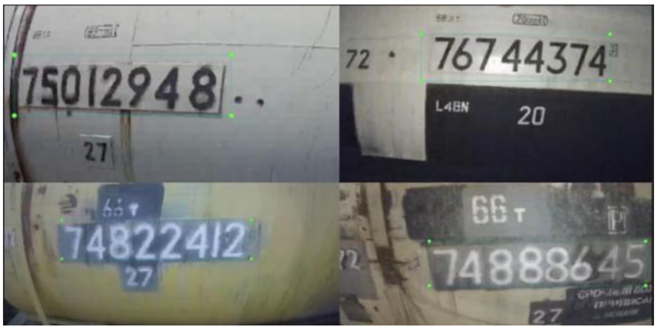


Fig 6: Number's borders

B. Number detection model

After the frame is classified, the number detection model is applied (Fig. 6). This model uses the ResNet-34 architecture. For this stage the ResNet-18 based model has a lot of errors, so we decided to grab a bigger model. We use the information from the classifier stage and try to find the number's border only on images where the wagon exists.

The main problem on this stage is to detect the number's border as accurately as possible. Sometimes there are some extra digits next to the number and the task of the number detector is to detect only the number content and get rid of extra artefacts next to the number. Using the detection model we simplify the task for our digit detection model to focus only on specific digit features.

C. Digit recognition model

If the number's border was detected on the image, the third model receives cropped number image and tries to recognize digits (Fig. 5).

This model is built on the powerful ResNet-50 architecture which can achieve impressive results in the digit recognition task. This big model can recognize digits in a different light, and weather conditions, also allowing to recognize of partially erased digits.

After that recognized digits are written down from left to right, the received number is processed with correctness checking (using checksum), and if it is correct, the system outputs this number as a result.

When the entire train passes by the video camera, the system generates a report with recognized wagon numbers and sends it to a remote server. The report contains information about each passed wagon: the time the wagon was detected, the track on which the wagon was detected (in the presence of a large number of cameras at the station) and the number of the detected wagon. The operator can compare recognized wagon numbers with their original images and correct errors if any.

III. Datasets

One of the challenges in train wagon number recognition is absence of public datasets with train numbers. We used it to create our own dataset. This dataset contains video sequences taken under various lighting and weather conditions.

A. Scene classifier dataset

The dataset for the scene classifier model has 7000 background image, 12000 gaps images and 11000 wagon images. It contains wagons of different shapes, colors, images of day and night conditions. It is important to clearly separate the cases where the gap between the wagons is on the left or right side of the image, because these transitive frames between the state "wagon" and the state "gap" have the highest probability of misclassification.

B. Number detector dataset

The dataset for the number detection model (Fig. 6) has 3400 wagon images with markings for the position of the number border. As shown in Fig. 6, next to the main number there are extra digits, so it is important to have a markup where the borders of the numbers will be highlighted as accurately as possible.

C. Digit recognizer dataset

The dataset for the digit recognition model (Fig. 5) has 3400 images where each digit is labeled. It includes numbers in different lighting conditions and with varying degrees of distortion. Fig. 7 shows digits count distribution in the dataset. This distribution shows the relative frequency of occurrence of digits in wagon numbers.

D. Validation dataset

We have a validation dataset with 2000 images of 500 different wagons to test number detection and digit recognition pipeline. There are 2-5 images per wagon, so this dataset can be used for number recognition pipeline validation. In real video processing we have multiple images of the same wagon, so if we want to test the recognition pipeline under similar conditions as it would in a real task, we should use a dataset with multiple wagon images and check if there was at least one correct recognition of the wagon number.

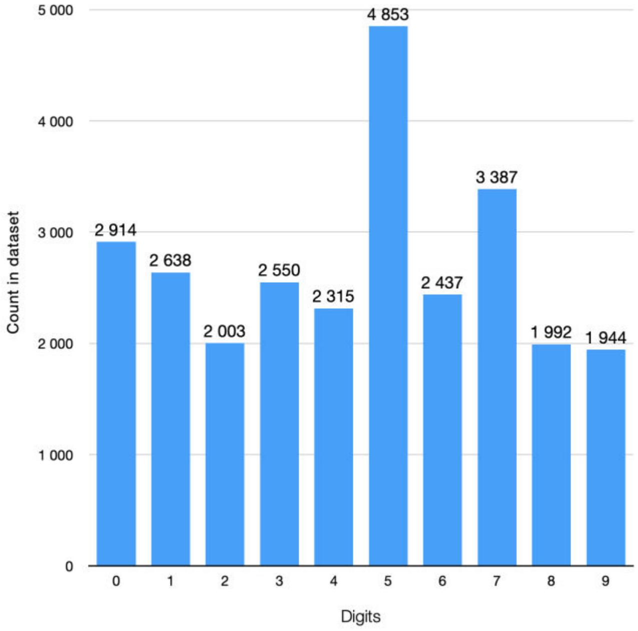
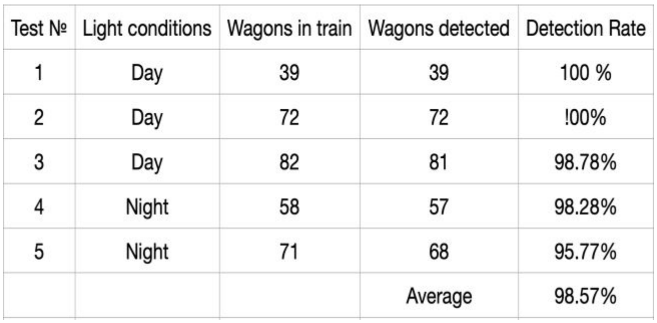


Fig 7: Digits count distribution in the dataset



Test №	Light conditions	Wagons in train	Wagons detected	Detection Rate
1	Day	39	39	100 %
2	Day	72	72	!00%
3	Day	82	81	98.78%
4	Night	58	57	98.28%
5	Night	71	68	95.77%
			Average	98.57%

Table 2: Wagon Counting Results

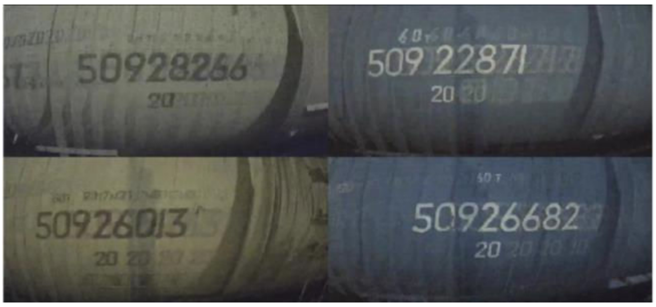


Fig 8 Examples of camera artifacts at night

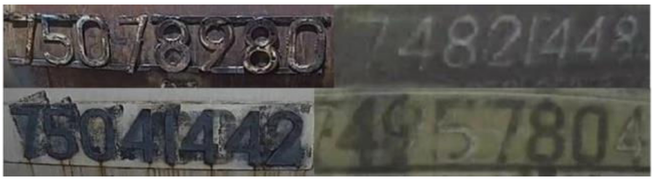


Fig 9: Examples of distortions (cropped numbers)

IV. Experiments

In this section, we describe the experiments carried out to validate the wagon count and number recognition.

A. Wagon Counting

Our solution is based on the use of only computer vision, unlike other solutions that use additional devices or sensors (for example, the approach proposed in [8]). Thus, implementation and maintenance costs are significantly lower.

The process of counting wagons is based on the scene classifier model. We can count wagons by analyzing the contents of the frame: if there are several wagon frames and after them several gap frames, we can increase a wagon counter by 1. There are also other solutions (the paper [2] introduces the wagon counting method based on tracking number positions). But our approach is more sustained because we can count wagons even if the wagon's number wasn't detected.

Then we process all the frames in the video using this approach and get the number of wagons in this video. Of course, we need to set a threshold for detecting multiple frames of the same time in a row to get rid of classifier model errors (for example, we will only detect the "wagon" state only if there are more than 5 frames with "wagon" in a row)

We took some videos of passing trains to check the accuracy of the car count. The proposed approach works well in the daytime and counts wagons with high accuracy. In nighttime videos, the quality of wagon counting is somewhat lower, since at night the images are poorly lit and the probability of a classifier error is higher. We achieved an average accuracy of 98.57% over 5 tests under different conditions (Table 2).

B. Number detection and digit recognition

We used our public validation dataset [10] to evaluate number detection and digit recognition pipeline. It is better to validate these modules together, because our task is to recognise the number and only the correct operation of both modules can lead to good results.

We have a validation dataset with 2000 images of 500 different wagons. There are 1-5 images per wagon to test. Our approach has recognised 97.15% (Table 3) of numbers correctly.

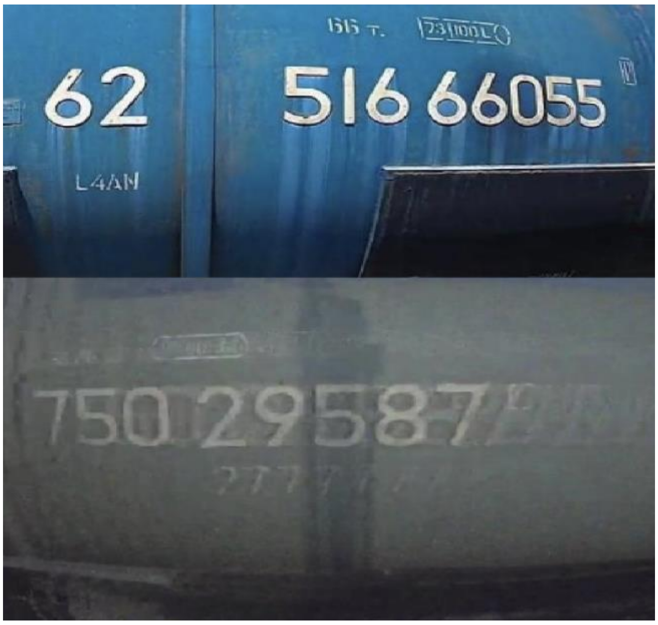


Fig 10: Comparison of daytime (top) and nighttime (bottom) conditions

Image count	Number recognized	Number not recognized
2000	1934	66

Table 3: Number Recognition Results (Images)

A more important metric is how many unique wagon numbers our solution recognised, and the results are quite good at 99.4% (Table 4). Since we have several number images of the wagon in real video processing (5 or more depending on the speed of the wagon), a more accurate estimation method is to predict several images of the same wagon and check if at least one number was correctly recognised.

Table 4: Number Recognition Results (Wagons)

Wagon count	Number recognized	Number not recognized
500	497	3

Our approach successfully handles images even with camera artefacts at night (Fig. 8). Our number detector has learned to detect only the number and avoid detecting mirrored numbers that appear due to lack of light. There are also a lot of examples with different distortions: dirt, erasure, rust (Fig. 9). These numbers are also successfully recognized with our approach.

During the experiments we found that if the number is not recognized, then it is likely to be damaged and not readable even by human. In other cases, when the wagon passes, the number is recognized at least once from a set of the wagon pictures, which is sufficient.

Fig. 10 shows a comparison between daytime and nighttime conditions. At night there may be too much distortion and the number may not be recognized, but this is a defect in the video recording since a person will not be able to recognize this number either.

V. Conclusion

In this work we presented efficient approaches for wagon counting and number recognition tasks. We achieved good results even under poor weather and light conditions. Our solution also works at night (compared to the solution described in [2]) and gets high accuracy. Compared with RFID-based methods, our approach has low installation and maintenance costs, while the accuracy of the recognition is about 99.4% using several images of the same wagon. Our solution can accurately recognize numbers even when a person cannot immediately identify them due to number distortion or poor lighting. Of course, the number may be unreadable due to damage or defects in the video, but in this case, even a person will not be able to recognize it.

Due to optimizations such as using a scene classifier, we achieve good processing speed even on CPU (Using 4 cores of Intel Xeon Silver 4210, our solution shows an average processing speed of 11 fps. It is sufficient for real-time video processing even without GPU usage.

In a future work, we intend to test different deep learning models to get even more accurate and faster solutions. We also plan to make an algorithm that will be able to predict the most possible number option if one digit is completely unreadable.

References

[1] W. Zhang, G. Zhou, and M. Jiang, "Convolutional Neural Network for Freight Train Information Recognition," in International Conference on Machine Learning and Computing (ICMLC), Feb 2017, p. 167–171.

[2] R. Laroca, A. C. Boslooper and D. Menotti, "Automatic Counting and Identification of Train Wagons Based on Computer Vision and Deep Learning," arXiv preprint arXiv:2010.16307, 2020

[3] Z. Liu, Z. Wang, and Y. Xing, "Wagon number recognition based on the YOLOv3 detector," in IEEE International Conference on Computer and Communication Engineering Technology, Aug 2019, pp. 159–163.

[4] X. Zou, Y. Fu, and X. Li, "Image feature recognition of railway truck based on machine learning," in IEEE Information Technology, Networking, Electronic and Automation Control Conference, Mar 2019, pp. 1549–1555.

[5] C. Li, S. Liu, Q. Xia, H. Wang, and H. Chen, "Automatic container code localization and recognition via an efficient code detector and sequence recognition," in IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2019, pp. 532–537.

[6] "Расшифровка номера вагона", [Deciphering the wagon number]. железнодорожник.рф https://железнодорожник.рф/poleznaja-informacija/rasshifrovka-nomera-vagona (accessed: July 5, 2021).

[7] A. Kumar, N. Ahuja, J. M. Hart, U. K. Visesh, P. J. Narayanan, and C. V. Jawahar, "A vision system for monitoring intermodal freight trains," in IEEE Workshop on Applications of Computer Vision, Feb 2007, pp. 24–30.

[8] D. Liya and L. Jilin, "Intelligent freight train ID recognition system," in IEEE International Conference on Intelligent Transportation Systems, Sep. 2002, pp. 417–422.

[9] "Знаки и надписи на вагонах грузового парка железных дорог колеи 1520мм" [Signs and inscriptions on the wagons of the freight fleet of 1520 mm gauge railways] pzdt.ru http://pzdt.ru/files/uploads/info/Знаки%20и%20надписи%20на%20вагонах%20грузового%20парка%20колеи%201520%20мм%20Альбом-справочник%20632-2011%20ПКБ%20ЦВ.pdf (accessed: July 5, 2021).

[10] "Dataset for wagon number recognition," 2022, EasyData, Available: https://www.easydata.nl/contact/resources/ (accessed: July 29, 2021).