

Robust TRISO-fueled Pebble Identification by Digit Recognition

Roshan Kenia¹, Jihane Mendil², Ahmed Jasim², Muthanna Al-Dahhan², and Zhaozheng Yin¹

¹Department of Computer Science, Stony Brook University

²Department of Chemical & Biochemical Engineering, Missouri University of Science and Technology

Abstract

Nuclear power plays a vital role in providing reliable and clean energy to fulfill increasing demands in electricity worldwide. It continues to be an essential source of national power supply as growing concerns about fossil fuel depletion, global warming, and emissions require utilizing sustainable energy sources. One area contributing to the growth of nuclear power is the development of reactors that have enhanced protection and security, thermal efficiency, and design. Reactor efficiency can be studied by the burnup that occurs when a TRISO-fueled pebble is inserted into the nuclear core and subsequently removed. The levels of burnup are measured based on the length of time the pebble spends within the core. In our design, each pebble is numbered by multiple digits printed in six locations using Ultra-High Temperature Ceramic paint. Naturally, computer vision techniques can be used to identify and time each pebble based on its digits as it enters and exits the core. We present a deep learning approach that successfully tags each pebble by identifying its digits from a video stream of the entrance and exit of the core. In a multi-step method, we extract only the clearest and most useful views of the pebble's digits to classify as it rolls by. This algorithm is robust against issues that occur for objects in movement such as motion blur, rotations, and glare. We outperform other state-of-the-art optical character recognition (OCR) models that fail to identify digits that are in motion. Our approach creates a safer and more efficient way to measure burnup within a core while contributing to the improvement of nuclear power produced by reactors.¹

1. Introduction

1.1. Pebble Bed Reactors

As one of the six classes of nuclear reactors being studied in the Generation IV initiative [14], pebble-bed reac-

¹Examples of pebble identification videos in supplemental materials.

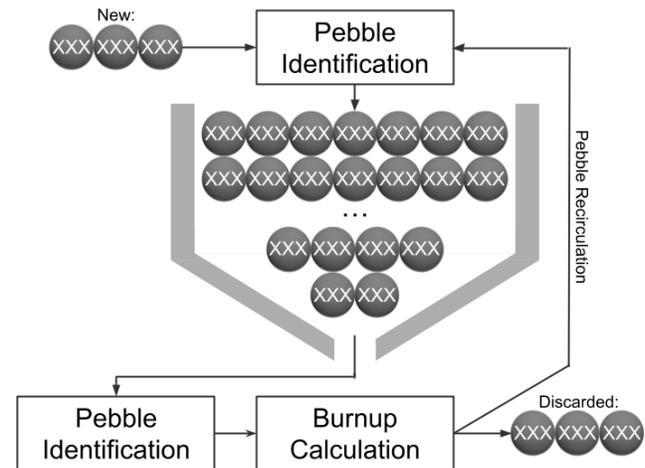


Figure 1. Pebble bed reactor that allows for continuous recycling to create energy. 'XXX': an arbitrary multi-digit number.

tors (PBRs) [13] have garnered lots of attention due to their safeness, modularity, and efficiency. They are an attractive solution to be used in the next generation of nuclear power production from their minimal environmental impact and reliability [13]. To power PBRs, tri-isotopic (TRISO) [25] nuclear fuel particles are mixed with graphite powder to create a spherical pebble 6 cm in diameter. About the size of a pool ball, these pebbles maintain their structural integrity when exposed to high temperature. This makes them the ideal candidate to be used for nuclear fission to take place.

The reactor core is filled with thousands of TRISO-fueled pebbles that are continuously added and recirculated as they undergo a process known as burnup [3] to fuel the system. During the refueling process, each pebble is removed one by one from the outlet of the reactor and measured for their accumulation of burnup. If the level of burnup is below a certain threshold, the pebble is sent back into the reactor. Otherwise, it is discarded as it cannot provide further fuel. The pebble recirculation as seen in Fig. 1 allows for more efficient nuclear energy production. It is an

advantage PBRs have over other nuclear core designs, as they do not need to be shut down for refueling.

1.2. Pebble Identification

It is crucial that these pebbles are not kept in the nuclear reactor core for a prolonged period as excessive burnup accumulation can occur which can have adverse effects on power production. Preventing this is a difficult task as a pebble's burnup can only be measured by the amount of time it has spent within the core. To accomplish this, each pebble must be identified as it enters and exits the core so that it can be properly timed. Appropriate pebble identification techniques need to be developed for PBRs to keep their design safe and efficient for maximal scale-up.

A simple way to uniquely tag each pebble is to give it a multi-digit number that can be painted on using Ultra-High Temperature Ceramic paint [24]. The numbers are visible after the pebble is passed through the reactor as the paint is resistant to nuclear core conditions such as radiation and high temperature. To cover all views, this number is placed in various locations on the pebble. This number can then be used to identify each pebble while it cycles through the refueling process for PBRs.

1.3. Problem Definition

As these pebbles are continuously fed in and removed out of the reactors, they must be identified to inhibit excessive fuel burnup or early stopping of fuel discharge. Suppose we have a pebble that is last seen at time t_1 from the video stream of the inlet of the reactor, as seen in Fig. 2. This pebble enters the reactor, spends some time within it, and then exits the reactor and can be seen at time t_2 from the video stream of the outlet of the reactor. The residence time t_r of this pebble can then be calculated as $t_r = t_2 - t_1$. Since each pebble is marked with a multi-digit number, we can match the timing of when it entered and exited the reactor. The task is to automatically identify and note the timing of the pebble twice: as it goes into the inlet and as it exits the outlet. This removes any need for a human to calculate the residence times manually. We are interested in utilizing computer vision techniques to ensure the automation of pebble identification is as accurate as possible.

1.4. Summary

With this nuclear engineering goal in mind, we present the following as our contributions:

- A novel approach to TRISO-fueled pebble identification that utilizes a new ceramic paint tagging system.
- The first computer vision-based system to accurately identify nuclear pebbles while they are in motion and rotating.

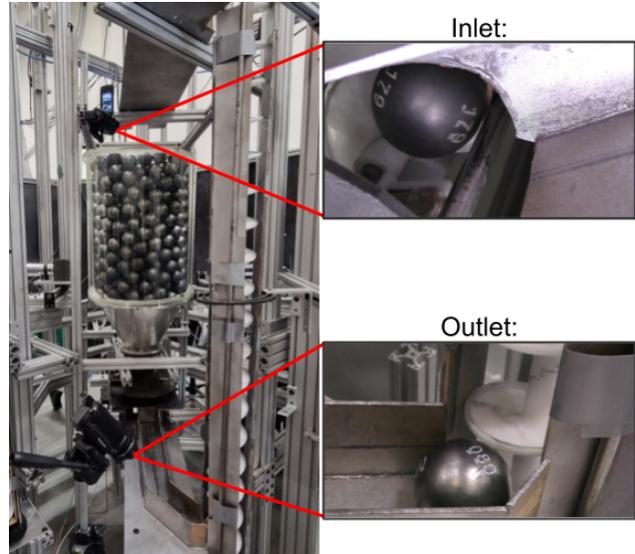


Figure 2. Example of pebble identification setup on the left. One camera is facing the inlet of the reactor (top). Another camera is facing the outlet of the reactor (bottom). Examples of numbered pebbles on the right.

- A fast and non-invasive way to calculate the residence time of TRISO-fueled pebbles, allowing the production of safer and more efficient Generation IV reactors.

2. Related Work

Identifying pebbles by X-ray imaging. The automatic identification and tracking of the pebbles that fuel the power generation within a nuclear core is a feature of PBRs that has not been extensively researched. One physical property of the pebbles that has been used in automatic recognition is the underlying TRISO particle distribution. Fang and Fulvio [4] show that pebbles can be uniquely identified inside the core by the three dimensional distribution of TRISO particles. They make use of X-ray CT scans of each pebble to reconstruct a three-dimensional distribution and match it to the corresponding pebble by aligning the two using the Go-ICP algorithm [32]. Similarly, Kwapis *et al.* [15] use a combination of X-ray imaging and deep learning to learn the TRISO particle distribution actively as new pebbles are introduced to the reactor as a way to identify the pebbles later on for burnup calculation.

The large amounts of gamma radiation being emitted from the irradiated fuel after use can have an impact on the pebble, *i.e.*, X-ray imaging might not always work as the particle structure is not guaranteed to be intact. In addition, these methods have been tested on synthetically generated small-scale pebble sets, and their application in a real-world scenario might not be practical. It requires the memorization of thousands of particles for thousands of pebbles,

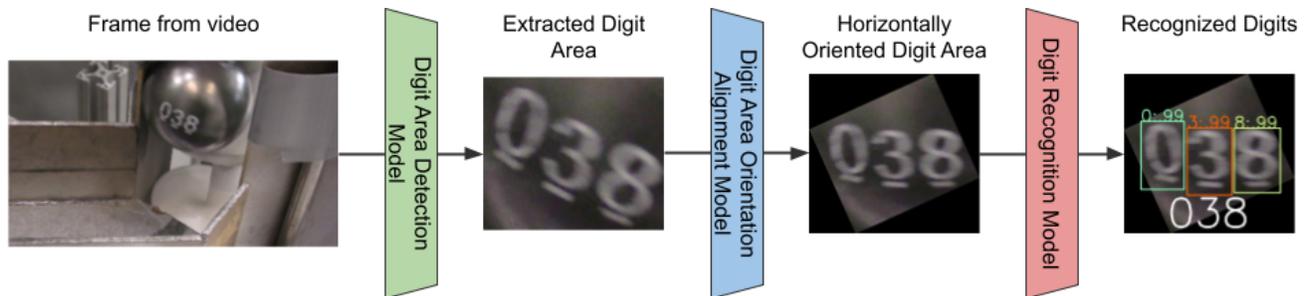


Figure 3. Overview of pebble identification pipeline.

which would not scale well. Alternatively, painting digits on the pebble surface using materials resistant to the heat inside a nuclear core is a promising scale-up solution.

Digit detection and recognition. The broad field of digit detection and recognition has had many works focused on solving different problems occurring in the real world. OCR engines like Tesseract [28], EasyOCR [11], Amazon Textract [27], Google Cloud Vision API [30], and ABBYY FineReader [1] have been developed to extract handwritten, typed, and printed text from documents and images. These methods act as a way to digitize the information presented, so that it can be used in a variety of tasks such as translation, text-to-speech, and summarization. There have also been multiple widely known and openly available digit datasets that have been used as another form of developing digit detection and recognition models. MNIST [16] contains a large amount of hand-written single-digits that can be predicted on with extremely high accuracy using recent methodologies [2,9,22]. In addition, the SVHN dataset [23] contains real-world images of multi-digits that represent the harder case of recognizing numbers within a natural scene.

Challenges and needs for digit detection and recognition on pebbles. In literature, there have been very few works addressing the issue of recognizing digits that are placed on moving spherical objects. General applications of this could be used in pool ball tracking or lottery number drawings, but in our case, it can be used for TRISO-fueled pebble identification. These spherical objects are subject to arbitrary rotations that can make the digits appear sideways, upside down, or diagonal. In addition, the curved surface of pebbles introduces out-of-plane image warping, which can cause confusion over digits that appear similar such as 5 and 6 or 1 and 7. Ordinary digit detection and recognition methods will fail to perform well for classifying digits on spherical objects as they require completely flat surfaces like documents or signs. An added complexity of objects in motion creates more difficulty recognizing digits as they are subject to blur and fuzziness. This major difference between previous work for digit detection and recognition and our TRISO-fueled pebbles problem requires a new methodology to tackle these issues and maintain a high performance.

3. Methodology

Our overall goal is to extract the frames from the video stream that contain the most useful information to detect each pebble and identify it correctly. To accomplish this, we forward each frame through a multi-step approach, as seen in Fig. 3. First, we break down the frame to detect the digit area. Then, the digit area on the pebble is aligned horizontally. If at any point we realize a digit area is too blurry or heavily occluded, we automatically throw it out. Once we obtain just the aligned digits from the frame, we can classify using our digit recognition model and save the most confident digits as the identification results.

3.1. Digit Area Detection

Detecting the digit areas of a pebble in a frame can boost the identification system’s efficiency and accuracy. First, it enables rapid processing of ignoring empty frames (with no pebbles) and continuing on frames with moving pebbles. Second, it restricts the digit recognition module to perform only on digit areas rather than wasting computation on irrelevant frame regions or introducing false positives.

Each frame of the video stream is processed through this step to detect if digits are present or not using a Mask R-CNN [7] (implementation details found in Sec. 5). If digits are present, they are segmented as a box region from the frame and saved for later processing. Otherwise, the processing of the frame stops and we move on to the next one. In the case multiple digit areas are detected on a pebble as seen in Fig. 4, we save and process each separately.

Since the video stream contains hundreds of pebbles rolling by, we need to be able to associate multiple detections of the same pebble over time. If there is a pebble present, we use the center of the digit area detection as its location representation to track it in each frame. We link these related locations by making use of the Euclidean distance between a new point and each moving pebble’s previous points. If they are close enough, the points are linked and the corresponding pebble’s location is continuously tracked.

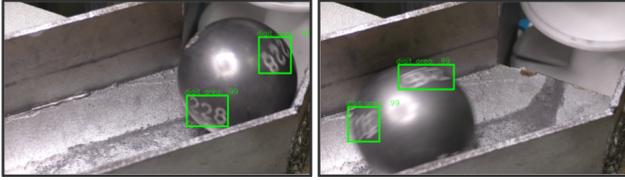


Figure 4. Example of digit area detection while the pebble is moving through the chamber.

3.2. Digit Area Orientation Alignment

Spherical pebbles in motion rotate while they are seen in each of the inlet/outlet view's video streams. Since the numbers for the pebbles are painted on, the digits will rotate as the pebbles rotate. It is difficult to correctly recognize digits that are randomly rotated as the digit's structural properties change. For example, digits such as 6 and 9 will be confused when rotated 180° , which will lead to incorrect recognitions being saved. Although methods like Spatial Transformer Networks [10] and Jain *et al.* [12] exist for rotated digit classification, they are either incapable or unable to properly perform on digits rotated more than -90° to 90° . Therefore, we must create our own method to solve this issue by utilizing our patterning method.

The numbers for the TRISO-fueled pebbles were painted with a bar underneath each digit to signify digit orientation. To align our digit area to a readable format, we must rotate the image to place the orientation bars in the bottom of the image. We first detect the three bars underneath each digit using a Mask R-CNN [7] (implementation details found in Sec. 5). We calculate the bounding box area's midpoint which roughly estimates to the middle digit's bar represented as point A. The center of the entire digit area image is represented as point B. Finally, to easily find the middle lower part of the image for angle calculations, we represent C as the point in the negative y direction from B that is equidistant as B to A. We calculate $\angle CBA$ as seen in Fig. 5 that represents the offset of the middle digit's bar from the middle lower part of the image.

Using the inverse angle, we apply an affine transformation [31] with a rotation matrix to place the middle bar in the image's middle lower region. This correction rotates the digit area of the pebble horizontally so that the digits are in the most readable state for recognition. If no bars were detected with confidence, we throw away this digit area crop and process any others for the frame. If no bars are detected with confidence for any crop, we move on to the next frame.

3.3. Digit Recognition

With correctly oriented digit areas, we can now recognize the digits present much easier than if we had not. The digits in the aligned digit area are recognized by a ten-class digit recognition model using a Mask R-CNN [7] (imple-

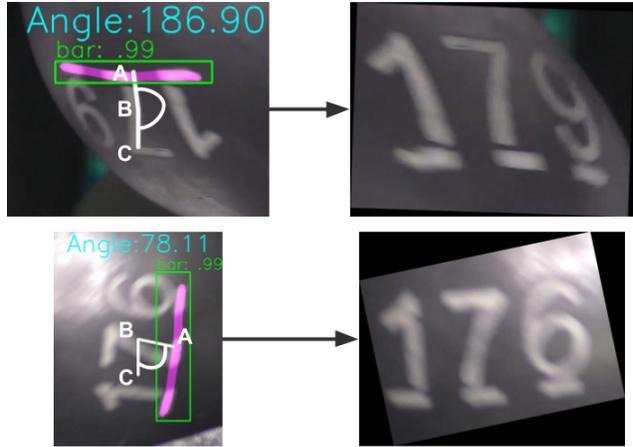


Figure 5. Example of digit area orientation alignment. We use the angle calculation to rotate the images on the left so that the digits can be horizontally read as seen on the right.

mentation details found in Sec. 5). We enforce the digit recognition inside the digit area to contain the most confident and distinct digits that have minimal overlap. The distinction between digits is calculated as the Euclidean distance between the midpoints of the bounding boxes of the classifications. Therefore, if one digit has multiple classifications, only the most confident one will be saved, similar to the nonmax suppression [26].

Since a pebble is in the video stream for multiple frames, we can make use of sequential recognition over a period of time to boost the recognition accuracy. Therefore, when a new digit area detection has a recognition output, we propose a voting system where the confidence score of each individual recognized digit of the pebble is accumulated. Then, once the pebble obtains a confident number of votes for every digit of a pebble, we can stop the repeated recognition of that pebble. The effectiveness of this recognition by sequential voting is validated in Sec. 6.2 and Fig. 7.

4. Datasets

As far as we know, there exists no other dataset of TRISO-fueled pebbles labeled with numbers. Thus, we created a benchmark dataset for this nuclear reactor application. In total, 600 pebbles were produced and labeled in 6 different locations with a 3 digit number using ceramic paint. A stationary photo of each pebble with their numbers perfectly aligned horizontally was saved. In addition, two training videos consisting of 14 and 21 pebbles respectively were captured moving through the outlet chamber. Our test set consists of one inlet chamber video and one outlet chamber video, each containing 43 and 47 pebbles respectively. Images and videos were captured using a Panasonic 4K Ultra HD Camcorder HC-VX981K. Video data was recorded

at a 1920x1080 resolution at 30FPS in a RGB format.

4.1. Datasets for Learning the Three Modules

Digit Area Detection Dataset. In order to complete our first task of detecting the digit area from the frame, we hand-labeled 652 frames selected from the two training videos. We labeled the digit area and saved it as a box. While training, we treated the box area as the digit area label and everything else in the frame as non-digit areas.

Bar Detection Dataset. Our second task of bar detection for digit area alignment also requires an object (bar) detection dataset. We selected 187 frames from the training videos and 192 randomly rotated stationary images and processed them through the digit area detection model to obtain the digit areas. These 379 digit areas contain the bars underneath the digits for constructing the bar detection dataset. The bars underneath the digits were annotated while everything else in the image was considered as non-bar areas.

Digit Recognition Dataset. First, we took the 600 stationary pebble images and used our model trained on the digit area detection dataset to obtain just the digit area. Then, we selected 142 frames from the training videos and similarly obtained just their digit areas. In both cases, we selected images that have digits that are nearly or perfectly horizontally aligned. Finally, we hand-labeled each digit in each digit area image using bounding boxes. There are 10 classes in total, representing digits from 0-9.

4.2. Data Augmentation

Due to the variations in nuclear reactor setups, we cannot always guarantee the same lighting and clarity. In order to account for this in the training of our models, we apply random data augmentations to each sample in each epoch.

First, we mimic blur associated with objects in motion by applying a Gaussian blurring [5] filter and then an affine transformation [31]. For Gaussian blurring, we randomly sample two positive odd numbers for the width and height of our kernel size, as a general blurring of the image. The affine transformation is then applied as a slight remapping of the image in one direction to act as a motion blur. We use a kernel size of 10x10 with a randomly sampled angle from 0° to 360° to apply this transformation.

Since we are working with spherical objects, rotations of the objects and digits are bound to happen while processing video frames. To help our models account for any rotation, we apply random rotations in each epoch to the images and their labels while training. To accomplish this, we randomly sample an angle from 0° to 360° (-5° to 5° for digit recognition) and apply an affine transformation [31] using a rotation matrix. Lastly, we use uniform random sampling to either decrease or increase the brightness of the image from 0.2 to 1.5 times the original. This can account for any changes in lighting for the video set-up.

Using these techniques, we can make use of our smaller datasets to learn robust models as no image will be exactly alike in any training iteration. In addition, the representations learned by the models are the most useful information rather than any artifact.

5. Implementation Details

For consistency and reproducibility, all three of our modules were accomplished using a Mask R-CNN [7] with a ResNet50 [8] backbone pre-trained on COCO [19] with a feature pyramid network [17]. We finetune the Faster R-CNN [6] classification head on our respective datasets and use only the bounding box output during testing. We found in our experiments, however, that the models within our methodology can be swapped with any object detection model and still perform well (comparison to [6, 18, 20, 29] in supplemental materials). In comparison to other major networks, the Mask R-CNN was less computationally expensive, which was attractive to use in a live setting along with its efficiency and high performance. We trained using Stochastic Gradient Descent with a momentum of 0.9 and a weight decay of 0.0005. We use Cosine Annealing Warm Restarts [21] to keep a varying learning rate while training. For all modules in both training and testing, we normalize the data before obtaining a prediction.

In our live implementations of our modules in Sec. 6.2, we make use of 100% of each respective dataset to train the three models for 200 epochs. However, to understand our first two individual model's performance by experiments in Sec. 6.1, we randomly sampled 75% of each respective dataset to act as a training set. After training for 100 epochs (due to smaller dataset), each model was tested on the rest of the 25% of their respective datasets. We repeated this experiment 5 times and obtained the mean and standard deviation for each distribution of bounding box scores. We examined the precision and recall at Intersection over Union (IoU) values of 0.50 and 0.75. To evaluate the third module (digit recognition) in Sec. 6.1, we use 100% of the digit recognition dataset to train and test on the two test videos.

6. Results

The proposed models are evaluated at two levels: the performance on individual modules and the identification accuracy on pebbles in video streams. Then, ablation studies are performed to evaluate the importance of each module.

6.1. Module Performance

Digit Area Detection. As seen in the first row of Tab. 1, we are able to obtain high precision and recall at both IoU levels. The frames of the videos are bound to change based on camera placement and therefore, could have completely different backgrounds than what is presented in the train-

Module	@IoU=0.50			@IoU=0.75		
	Precision	Recall	F-Score	Precision	Recall	F-Score
DAD	99.36 \pm 0.36	99.86 \pm 0.22	99.61 \pm 0.28	94.30 \pm 2.65	96.00 \pm 1.74	95.14 \pm 2.21
DAOA	97.94 \pm 0.46	99.48 \pm 0.39	98.70 \pm 0.41	93.94 \pm 1.75	96.22 \pm 1.21	95.07 \pm 1.49

Table 1. Results based on bounding box borders for our first two modules: Digit Area Detection (DAD) and Digit Area Orientation Alignment (DAOA). We calculate the precision, recall, and F-score values with Intersection over Union (IoU) thresholds of 0.50 and 0.75 for 5 trials. Afterwards, we average these values and calculate their standard deviation.

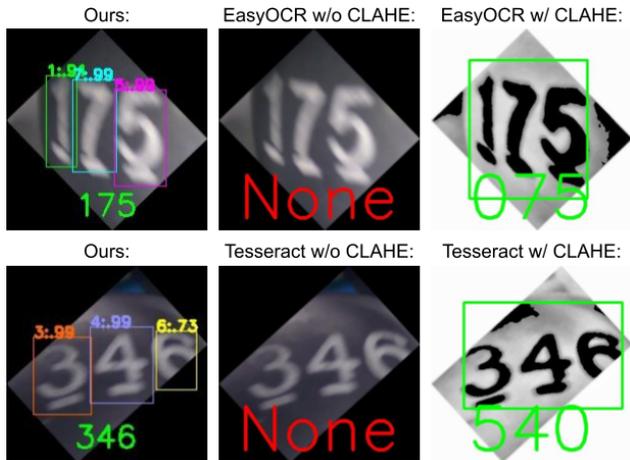


Figure 6. Left: Our method recognizes digits with high confidence. Middle: EasyOCR/Tesseract recognizes nothing on the pebble digits. Right: Using some image preprocessing, EasyOCR/Tesseract can recognize some digits on pebbles. Larger version found in supplemental materials.

ing videos. By including the negative areas as a label while training, this model contains far less false positive and false negative detections. In addition, glare is usually seen in a very concentrated area on the pebble, which makes it structurally different from digits. Our model is able to distinguish between the two from ample positive and negative labeling. The only difficulty that may arise is when the glare is concentrated on the digits of the pebble, but these instances generally do not have strong predictions in the next two modules and are therefore discarded.

Digit Area Orientation Alignment. Similar to the first module, our digit area orientation model is able to achieve high performance, as seen in the second row of Tab. 1. With accurate bar detections, we are able to calculate the angle to orient the digit area much easier.

Digit Recognition. There are many widely known digit datasets openly available to the public that we can use to strengthen our digit recognition module. We found the SVHN dataset to represent multi-digit numbers in the natural world similar to the digits on our TRISO-fueled pebbles. Therefore, we pretrained our digit recognition model on a random selection of 10,000 images from the SVHN training

set for 200 epochs. It was then finetuned on our multiclass digit dataset from the pebbles as described in Sec. 4.1.

There are a few OCRs, such as Tesseract and EasyOCR, that have the capability to predict on natural images with ease. However, we have to preprocess our aligned digit area detections to the format OCR models perform best by applying contrast-limited adaptive histogram equalization (CLAHE) [33] to all images. As seen in Fig. 6, this one preprocessing step formats the images in a way that appears similar to how text would be in a regular document. Afterwards, we processed these images through Tesseract and EasyOCR to obtain multi-digit number predictions that are then split into individual digit recognitions.

As we can see in the “Individual Digit” column of Tab. 2, our individual digit recognition model greatly outperforms the Tesseract and EasyOCR models with CLAHE. We believe the plethora of information learned about digit styles and shapes from the SVHN dataset helps our model adapt well to the digits of the pebbles. Figure 6 exemplifies how our method is robust to issues such as motion blur and cut-off digits in comparison to EasyOCR and Tesseract. Additionally, we believe the main issue in classifying on pebble images with an OCR model is the slight glare on the pebble that blends with the digits even after CLAHE.

6.2. Pebble Identification Performance

Although single-digit recognition performance was an important part of our pipeline, we wanted to understand how well it worked as a whole in pebble identification. As explained in Sec. 3.3, we can leverage knowledge gained over multiple frames of a video for a pebble, as seen in Fig. 7, in order to obtain the most confident digit recognition possible using a voting system. We tested our complete methodology on the two test videos. After the pebble has left the frame, we take the most probable digits for each digit of the multi-digit number based on our normalized accumulated voting scores and create the pebble identification classification. If the final number classification matched the three-digit number painted on the pebble, we classified it as correct. Otherwise, even if one digit was off, we classified it as incorrect.

We also compared the effect of using different digit recognition methods within our pipeline for pebble recognition. As we can see in the “Pebble Identification” column of Tab. 2, the model pretrained on SVHN and then

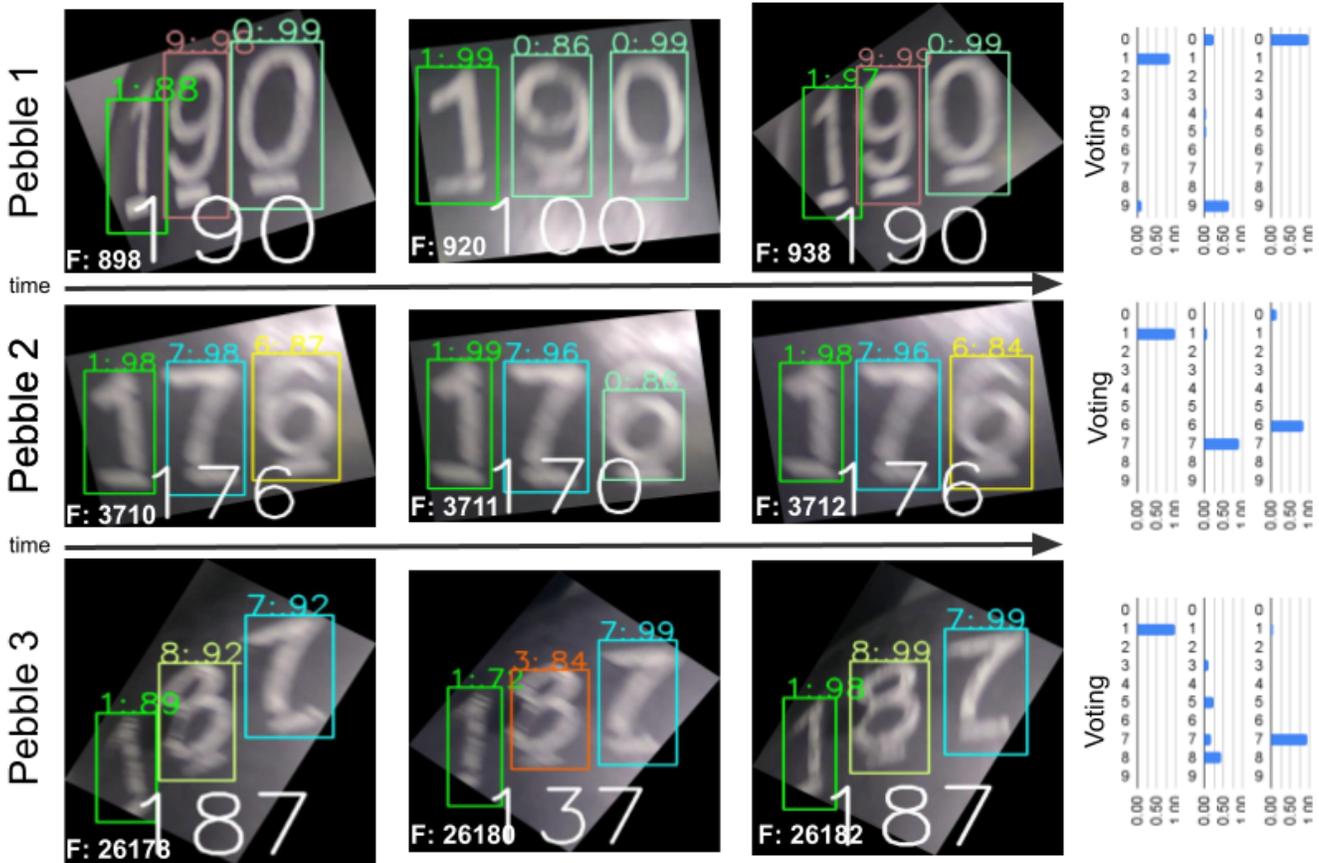


Figure 7. Example of pebbles being detected over a sequence of frames. A collection of predictions allows for slight errors to have little impact on the overall final classification. In the last column, we show the normalized ten-class scores for each digit of the multi-digit number. These scores represent the most probable digits that are concatenated and used in our overall pebble identification. We can see digits that have similar structures are confused the most. ‘F: X’ denotes the Xth frame. ‘1:0.88’ means the digit is recognized as ‘1’ with a 0.88 confidence score.

Method	Video 1 (47 Pebbles)		Video 2 (43 Pebbles)	
	Pebble Identification	Individual Digit	Pebble Identification	Individual Digit
Tesseract [28]	27/47 (57.45)	294/330 (89.09)	25/43 (58.14)	293/330 (88.79)
EasyOCR [11]	39/47 (82.98)	645/687 (93.89)	39/43 (90.70)	650/679 (95.73)
Ours	42/47 (89.36)	814/849 (95.88)	42/43 (97.67)	744/755 (98.54)
Ours w/o Orientation	3/47 (6.38)	144/331 (43.50)	14/43 (32.56)	286/457 (62.58)

Table 2. Results for two test videos. Pebble Identification accuracy represents the number of pebbles that were correctly identified within the entire video. Individual Digit accuracy denotes the number of correctly recognized single-digit over all detected digits (e.g., 294/330 means Tesseract only detects 330 single-digits and correctly recognizes 294 of them). The denominator number of individual digit recognitions is based on how many each method was able to detect with very high confidence.

finetuned on our multiclass digit dataset (Ours) obtains the best results for overall pebble classifications, compared to Tesseract and EasyOCR. We believe the extra knowledge gained from pretraining on SVHN allowed this model to make many more high confidence individual digit predictions. In some scenarios, digit recognitions had a weaker confidence resulting in an incorrect single-digit contribut-

ing to the overall pebble identification voting. However, by an increased weighting of the most confident individual digit recognitions, we can amend the small errors that may occur on some frames. These corrections can be seen in Fig. 7 contributing to the voting score probabilities and greatly improving overall pebble identification accuracy.

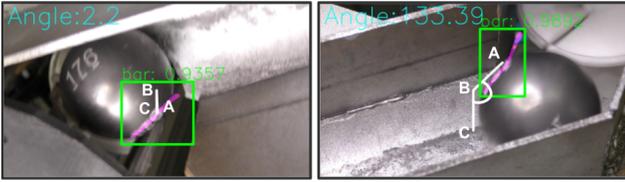


Figure 8. Example of digit area orientation alignment with no digit area detection.

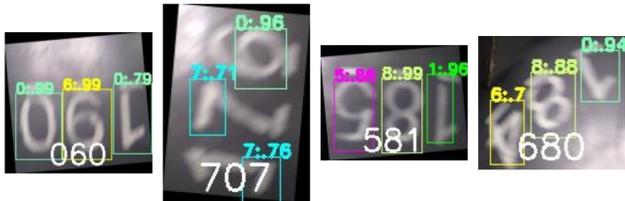


Figure 9. Example of how no digit orientation alignment leads to poor digit recognition and pebble identification.

6.3. Ablation Study

Our method contains three main parts: digit area detection, digit area orientation alignment, and digit recognition. The overall task is to classify the digits on the pebbles so the last module of digit recognition is a necessity. In the following, we evaluate the effectiveness of the other two modules.

The effectiveness of digit area detection. We removed our digit area detection module while keeping the second and third modules and attempted to classify the pebbles in the test videos. Therefore, every new frame skipped the first module and was instead fed to our second module to attempt aligning the digit areas. As we can see in Fig. 8, the second module fails to complete its task as the frame background and pebble digits are picked up as bars for alignment. Without properly aligned frames, we cannot accurately perform the digit recognition task (0 accuracy). It is therefore necessary for the digit areas to be detected as background noise will hinder the performance of any further steps.

The effectiveness of digit area alignment. We removed the digit area orientation alignment module from our pipeline while keeping the first and third modules and once again attempted to classify the pebbles in the test videos. Each frame had its digit area detected and cropped but was not oriented before being classified using our digit recogniser. As we can see in Fig. 9, the third module can be used on the unoriented digit areas, but many errors on individual digit recognition eventually impact the overall pebble voting system. First, our digit recogniser is trained to account for slight rotations, but it is difficult to confidently predict on completely sideways or upside down digits. Second, the digit area orientation alignment module works as a method to remove extremely blurry images as the bars underneath the digits will not be detected. The removal of this align-



Figure 10. Failure cases. 10a and 10b represent failed digit area detections. 10c represents a misclassification on similar digits.

ment module leads to an increase in bad digit recognitions as blurry images are classified by the digit recogniser. These two effects ultimately result in a much worse pebble identification accuracy as noted in the last row of Tab. 2.

6.4. Failure Cases

The few errors in overall pebble identification our methodology has are mainly due to the speed at which the pebbles roll by in the video data. When the pebbles move too fast, extreme motion blur makes it difficult to obtain clear digit areas to orient and classify, as seen in Fig. 10a. This can be increasingly difficult combined with the severe glare as seen in Fig. 10b. In addition, digits that are very similar in shape such as 5 and 6 or 1 and 7 can be confused and contribute to an incorrect pebble identification as seen in Fig. 10c. Although our continuous pebble identification by voting, as illustrated in Fig. 7, can overcome errors in some frames, the identification pipeline may fail if the misclassification or false negative due to extreme blur and glare dominates the pebble's entire existence in the video stream. These issues are not simply resolved as they would require alterations of the reactor set-up and pebble patterning.

7. Conclusion

We present a holistic approach to TRISO-fueled pebble identification using a unique ceramic paint digit patterning system. Utilizing our digit area detection and orientation alignment methods, our identifications are robust to issues that would normally happen to pebbles rolling by in a video stream such as motion blur and rotations. We outperform widely known OCR models that struggle on digits that may be warped when placed on spherical objects. In addition, creating brand new datasets allows us to adapt a model pre-trained on SVHN to perform well on our pebble data. Even with smaller amounts of data, our method will scale up well with hundreds of thousands of pebbles for nuclear energy production. We also do not require the use of X-rays and can rely on the surface patterning to remain intact through the core conditions due to its heat and abrasion resistant properties. This work aims to lead to a new line of TRISO-fueled pebble identification methods that can be researched and implemented in the next generation of nuclear reactors.

References

- [1] ABBYY. Abbyy finereader pdf. <https://pdf.abbyy.com/>. 3
- [2] Adam Byerly, Tatiana Kalganova, and Ian Dear. No routing needed between capsules. *Neurocomputing*, 463:545–553, 2021. 3
- [3] Xue-Nong Chen, Edgar Kiefhaber, and Werner Maschek. Fundamental burn-up mode in a pebble-bed type reactor. *Progress in Nuclear Energy*, 50(2-6):219–224, 2008. 1
- [4] Ming Fang and Angela Di Fulvio. Algorithms for triso fuel identification based on x-ray ct validated on tungsten-carbide compacts. *arXiv preprint arXiv:2204.13774*, 2022. 2
- [5] Estevão S Gedraite and Murielle Hadad. Investigation on the effect of a gaussian blur in image filtering and segmentation. In *Proceedings ELMAR-2011*, pages 393–396. IEEE, 2011. 5
- [6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5
- [7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 3, 4, 5
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [9] Daiki Hirata and Norikazu Takahashi. Ensemble learning in cnn augmented with fully connected subnetworks. *arXiv preprint arXiv:2003.08562*, 2020. 3
- [10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks, 2016. 4
- [11] JaidedAI. Easyocr. <https://github.com/JaidedAI/EasyOCR>. 3, 7
- [12] Ayushi Jain, Gorthi R. K. Sai Subrahmanyam, and Deepak Mishra. Rotation invariant digit recognition using convolutional neural network. In Bidyut B. Chaudhuri, Mohan S. Kankanhalli, and Balasubramanian Raman, editors, *Proceedings of 2nd International Conference on Computer Vision & Image Processing*, pages 91–102, Singapore, 2018. Springer Singapore. 4
- [13] Andrew C Kadak. A future for nuclear energy: pebble bed reactors. *International journal of critical infrastructures*, 1(4):330–345, 2005. 1
- [14] John E Kelly. Generation iv international forum: A decade of progress through international cooperation. *Progress in Nuclear Energy*, 77:240–246, 2014. 1
- [15] Emily H Kwapis, Hongcheng Liu, and Kyle C Hartig. Tracking of individual triso-fueled pebbles through the application of x-ray imaging with deep metric learning. *Progress in Nuclear Energy*, 140:103913, 2021. 2
- [16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3
- [17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 5
- [18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 5
- [19] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 5
- [20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015. 5
- [21] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5
- [22] Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: Capsule network with self-attention routing. *Scientific reports*, 11(1):14634, 2021. 3
- [23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 3
- [24] Dewei Ni, Yuan Cheng, Jiaping Zhang, Ji-Xuan Liu, Ji Zou, Bowen Chen, Haoyang Wu, Hejun Li, Shaoming Dong, Jiecai Han, et al. Advances in ultra-high temperature ceramics, composites, and coatings. *Journal of Advanced Ceramics*, 11:1–56, 2022. 2
- [25] Jeffrey J Powers and Brian D Wirth. A review of triso fuel performance models. *Journal of Nuclear Materials*, 405(1):74–82, 2010. 1
- [26] Azriel Rosenfeld and Mark Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 100(5):562–569, 1971. 4
- [27] Amazon Web Services. Amazon textract. <https://aws.amazon.com/textract/ocr/>. 3
- [28] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007. 3, 7
- [29] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: A simple and strong anchor-free object detector, 2020. 5
- [30] Jake Walker, Yasuhisa Fujii, and Ashok C Popat. A web-based ocr service for documents. In *Proceedings of the 13th IAPR international workshop on document analysis systems (DAS), Vienna, Austria*, volume 1, 2018. 3
- [31] Eric W Weisstein. Affine transformation. <https://mathworld.wolfram.com/>, 2004. 4, 5
- [32] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015. 2
- [33] Karel Zuiderveld. Contrast limited adaptive histogram equalization. *Graphics gems*, pages 474–485, 1994. 6