

A Deep-Learning Approach to Marble-Burying Quantification: Image Segmentation of Marbles and Bedding

Yicheng Zhu, Brandon Hudson, Chandranil Chakrabortii, Yun-Hsuan Su, and Kevin Huang

Abstract—This paper presents and evaluates three automated tools for semantically segmenting images from marble-burying experiments. The marble-burying animal model is widely used as an indication of anxiety or obsessive compulsive behavior in rodents. In general, the tendency for caged rodents to bury objects in their bedding is seen as anxiety related, and several methods have been proposed to measure the degree of this burying behavior. Unfortunately, most of these methods are coarse or require either subjective interpretation or onerous manual procedures. Digital imaging can provide pre- and post-experiment burying states as well as a platform for a standardized and streamlined quantification of the marble-burying test. While continuous imaging streams might provide more information and temporal analysis, such datasets are rare, require expert annotation, and can be prohibitively large. The authors propose that single-image semantic, pixel-wise segmentation of marble and bedding pixels are key components that can enable effective quantification. For example, the ratio of marble to bedding pixels can provide greater granularity in assessing marble-burying behavior. In this work, a classical image segmentation approach, a single-class U-Net and a multi-class U-Net were comparatively evaluated via standard segmentation metrics. Results show that the deep-learning methods demonstrate greater segmentation performance than the traditional method. Timing-performance trade off considerations between single- and multi-class methods are also explored.

Index Terms—U-Net; image segmentation; automation; machine vision; marble-burying; open-source tool

I. INTRODUCTION

The burying behavior as a defensive mechanism of rodents in laboratory settings was first observed and introduced by Pinel and Treit [1]. This was in contrast to the natural defensive reactions of other lab test animals that typically exhibited freezing, fleeing or attacking responses. Since then, the marble-burying test has been used for both pharmacological and behavioral testing [2]. A typical pair of pre- and post-treatment marble-burying experiment images are shown in Figure 1. However, a standard, automatic and quantifiable

measure of marble-burying behavior in rodents has yet to be widely deployed. The work presented here leverages digital imaging and machine learning techniques to automatically segment semantically relevant portions of marble-burying image data with promising results. Figure 2 depicts the overall experimental procedure.

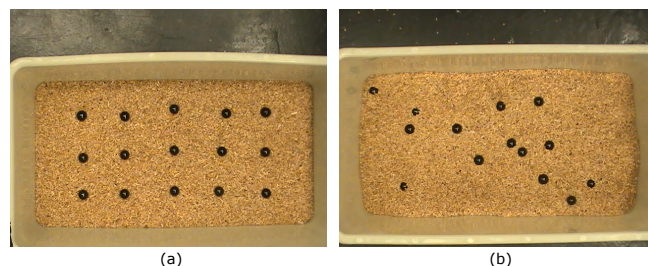


Fig. 1. Example vertical images (a) pre-experiment (b) post-experiment marble and bedding state. The test subject mouse is allowed to bury marbles between states (a) and (b), and the degree of burying activity is seen to correlate with anxiety or obsessive compulsive behavior.

A. Related Work

1) **Marble-Burying:** Marble-burying is frequently used as a behavioral assessment and pharmacological tool. Prominently, it is one of several behavioral indicators for measuring anxiety, anti-anxiety effects, and stress inducing or reducing treatments [3]. The model is based on the observation that upon encountering foreign or strange objects in a laboratory setting, mice exhibit distinctive burrowing, digging, hoarding and burying behavior [4]. Other tests for evaluation of anti-anxiety drugs include elevated plus maze, light dark box, open-field test, and the hole-board test [2]. Taylor et al. found that marble-burying behavior is related to γ -Aminobutyric acid, or GABA, and serotonin, and could thus be useful in pharmacological drug tests for attention deficit and impulsivity [5].

Several factors may affect stress and thus marble-burying behavior, such as lighting and temperature. In practice, other lab-setting considerations - such as consistent sterilization [6] and depth of bedding [7] protocols, should be maintained. A typical marble-burying setup contains standard size polycarbonate cages with sterilized, unscented bedding at a depth of 5 cm. The standard procedure is as follows: a fixed number of identical marbles are positioned uniformly on the bedding before the test animal is placed in the cage. After a set time period (30 minutes), the animal is removed and the total number of buried marbles is recorded, where only marbles buried ≥ 3 or more are counted towards the number of buried marbles [7], [8].

This material was supported by the Trinity College Summer Research Program and is based upon work supported by the National Science Foundation under Grant No. IIS-2101107. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF. The authors would like to thank David Ruskin and Susan Masino from Trinity College for providing raw image data sets.

Yicheng Zhu, Brandon Hudson and Kevin Huang are with Trinity College, Dept. of Engineering, 300 Summit St., Hartford, CT 06106 USA {yicheng.zhu, brandon.hudson, kevin.huang}@trincoll.edu

Chandranil Chakrabortii is with Trinity College, Dept. of Computer Science, 300 Summit St., Hartford, CT 06106 USA nil.chakrabortii@trincoll.edu

Yun-Hsuan Su is with Mount Holyoke College, Department of Computer Science, 50 College St, South Hadley, MA 01075 USA msu@mtholyoke.edu

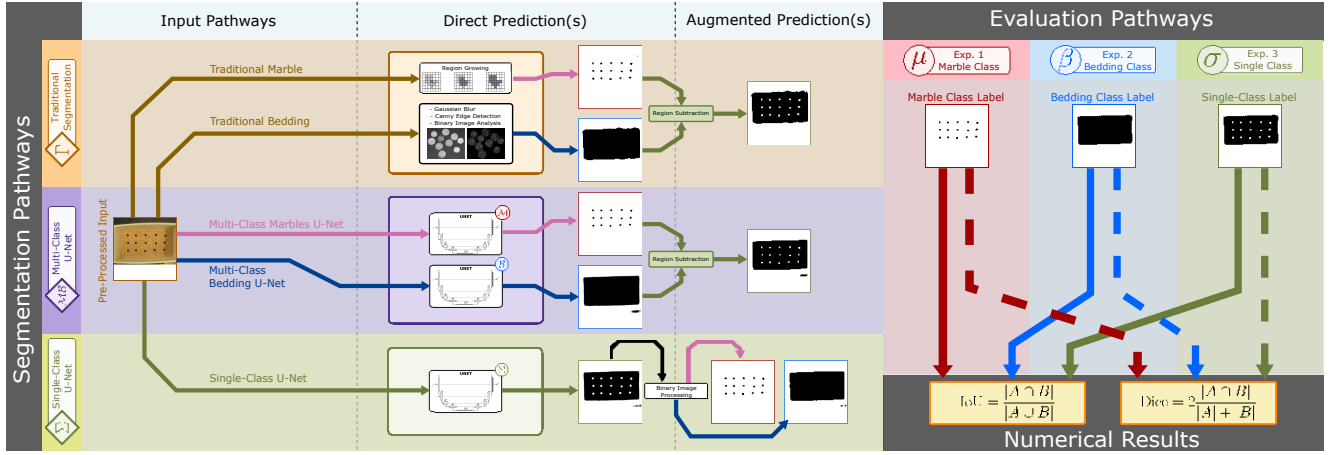


Fig. 2. Experimental workflow for evaluation. Each input image undergoes three separate segmentation pathways: Γ – a traditional image segmentation method; $M B$ – a multi-class U-Net segmentation network; Σ – a single class U-Net segmentation network. Each pathway generates one prediction of each of three classes: μ – marble class; β – bedding class; σ – single-class. The generated predictions are then evaluated against the three class labels.

2) Towards Better Quantification: The notion of “2/3 or more buried” is subjective and difficult to replicate or standardize. Instead, non-objective and image based approaches are sought to provide a more consistent basis for marble-burying measurement. Towards that end, photographs pre- and post-test can be manually assessed [9], [10]. Similarly, in a study examining the effects of ketogenic diets on mice [11], a digital camera rig was used to capture vertically from above images of marble-burying experiments to evaluate pain (examples in Fig. 1 were obtained from those experiments) [12]. Using Adobe Photoshop, black marble pixels and bedding perimeter shape were then manually extracted. The final measure of marble burying was proposed as the number of marble pixels as a percentage of the entire burying field’s pixels. This method provides more objectivity and granularity to burying behavior, yet requires copious amounts of manual image processing. In an approach to automate and standardize marble burying tests, Wahl et al. [13] sought a supervised machine learning approach to examine in detail both the temporal and spatial behavior of the test animals. For this, inputs were video streams of marble-burying tests. These data were processed in Mathworks MATLAB and Janelia Automatic Animal Behavior Annotator. While the method provides automation and introduces temporal analysis of the actual burying behavior, video datasets and manual annotation of training data (pixels coupled with activity duration) can be both time consuming and potentially subjective. [13] required experts for repeatable labeling of mouse behavior duration. The less data intensive and more readily reproducible labeling of image pixels is more accessible, and is thus the modality that the authors seek to explore.

B. Image Segmentation

Classical image segmentation techniques combine digital image processing and optimization methods. They can largely be broken down into layer-based and block-based, the latter of which can focus on region- or edge-based approaches [14]. Most of these traditional methods rely heavily on heuristics and are not generalizable. Deep learning methods seek to overcome these drawbacks. Most semantic

image segmentation networks utilize encoder decoder architectures. Take, for example, Efficient Residual Factorized Convolutional Network for Real-Time (ERFNet), which incorporated residual connections and factorized convolutions to decrease computational cost [15]. Takikawa et al. [16] introduced Gated Shape Convolutional Neural Networks (Gated SCNN), which implement a parallel pathway for computing wire shapes near boundaries in conjunction with a deeper architecture for larger objects. Both have found success in intelligent vehicle semantic scene segmentation.

In this work, the authors focus on the U-Net [17] architecture. The U-Net is a deeply-supervised and symmetric encoder-decoder network that leverages skip connections. It is amenable to feature enhancement [18], [19], incorporation in generative adversarial networks [20], and has found particular popularity in biomedical image segmentation. For example, the U-Net has been used for volumetric segmentation of tumors from tomography [21], [22], as well as for tool segmentation in robot-assisted minimally invasive surgery [23], [24]. The U-Net was chosen in this work because of its recognition and widespread popularity and accessibility, as well as its efficiency at solving pixel-wise segmentation.

C. Contributions

This paper presents and evaluates methods of segmenting both marble and bedding regions from marble-burying images. To the best of the authors’ knowledge, this work is the first to simultaneously present:

- i) a pre-processing and annotation protocol for vertical marble-burying images;
- ii) three separate methods for marble-burying image segmentation:
 - Γ – a traditional image segmentation method;
 - $M B$ – a multi-class U-Net segmentation network;
 - Σ – a single class U-Net segmentation network;
- iii) a comparative analysis of segmentation performance on multiple semantic classes coupled with timing and performance considerations;
- iv) an open source code base [25].

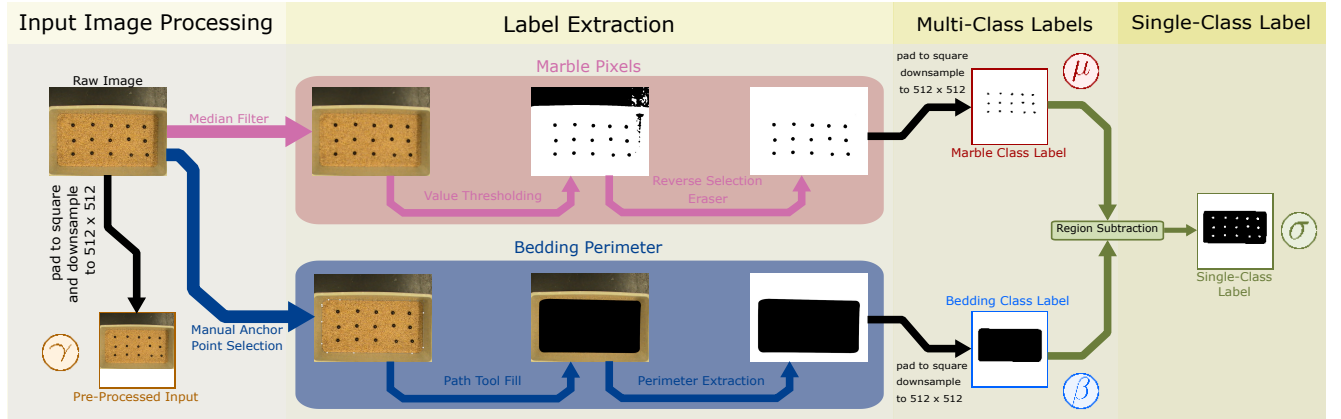


Fig. 3. Image processing for input dataset creation as well as label creation for both multi-class and single-class U-Net segmentation networks. Each image within the entire 526 image dataset undergoes this process to generate four separate 512×512 experiment-ready images: γ – pre-processed input; μ – marble class annotation; β – bedding class annotation; σ – single-class label.

II. METHODS

The components that enable the system workflow as depicted in Fig. 2 are described below. The procedural methodology for this research consisted of dataset pre-processing, annotation preparation, segmentation methods, and evaluation procedures.

A. Dataset

1) **Raw Data:** The animal studies from which the data were collected were performed in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals. The experiments were furthermore approved by the Institutional Animal Care and Use Committee of Trinity College (A3869-01). The cages used were 19×29×13 cm in size. For each experiment, 15 black 1.6 cm diameter marbles were arranged in a 3×5 atop 5 cm of wood ship bedding [12]. Each experiment lasted 30 minutes, and a pair of pre- and post-test images were taken per experiment using a digital camera (different cameras were sometimes used to capture data, resulting in different resolutions including 640×480 and 2576×1932 pixels). Thus, over the 213 trials, a total of 526 images were collected and form the dataset used in this work. Experimental lighting conditions and camera positioning varied slightly between trials.

2) **Pre-Processing and Annotation:** Each image in the dataset underwent the same process to create an element in the following sets:

- γ – pre-processed input;
- multi-class labels:
 - μ – marble class annotation;
 - β – bedding class annotation;
- σ – single-class label.

a) **Input Images, γ :** Since the raw data varied in sizes, lighting, and orientation, the images were first processed into the same format. All images were first converted to RGB color images with bit depth of 24 bits. The resultant images were augmented by padding white pixels on the shorter dimension until square. Finally, the image was uniformly downsampled to be 512×512. Color space and encoding

manipulations were performed with image processing methods and manual tools using the GNU Image Manipulation Program (GIMP 2.10.28) and pixel-level matrix calculations in Mathworks MATLAB (R2021-b, version 9.11.0.1873467). This is summarized in the leftmost column of Fig. 3.

b) **Multi-Class Labels:** The multi-class U-Net aims to segment two separate semantic categories, namely marble pixels and bedding pixels. To that end, each image requires two accompanying annotations, one for each semantic group.

Marble Class Annotation, μ , is depicted in the upper pathway in Fig. 3. To extract the marble pixels from the raw image, first high frequency artifacts, such as reflections, are smoothed with a median-blur filter. The filter is chosen to be a circular window with radius of three pixels, with target percentile for both color and alpha channels being the 50th percentile. To extract the marble borders and retain in a black-and-white image, a binary threshold on the value channel in HSV space is tuned for each image. Subsequently, the non-marble pixels that were not eliminated via thresholding are not connected to marble pixels. Thus, they are simply manually erased using the reverse selection tool. This image is then padded with white pixels along the shorter dimension until square, then down-sampled to 512×512.

Bedding Class Annotation, β , is depicted in the lower pathway in Fig. 3. The bedding often exhibits reflections in the polycarbonate cage walls. Thus, simple image processing tools were found to not reliably assist in bedding annotation. Instead, path nodes were manually selected in each raw image to form the vertices of a polygon that follows closely the bedding perimeter. The result was padded to square with white pixels then down-sampled to 512×512.

c) **Single-Class Label, σ :** Since the marble annotation is within the bedding annotation for each image, a simple subtraction of the multi-class annotation intersections from the bedding annotation forms the single-class annotation. This is depicted in the rightmost column of Fig. 3. Thus this single class encodes information both about marbles and bedding outline for each image.

B. Segmentation Procedure

With raw data pre-processed to consistent format and three annotation classes created, the segmentation pathways can be developed or trained. In total, three segmentation pathways are built and tested, see Fig. 2. All computational processes were conducted on single desktop machine.

1) System Hardware and Codebase: The training and implementation of segmentation methods were carried out on a machine equipped with an AMD Ryzen 7 3700x 8-core processor with 16-GB DDR4 RAM, NVIDIA GeForce RTX 2060 graphics card, and running Ubuntu 20.04.3, 64-bit operating system. Training and testing were performed under hardware acceleration with GPU-runtime catered to the specifications of this machine and graphics card - while graphics acceleration is not necessary, training speed is significantly improved. The standard TensorFlow package without acceleration is also suitable, however timing considerations presented in this work reflect performance with graphics acceleration. The python environment setup file for this work (.yml filetype extension) has been uploaded to the authors' repository [25] and is open for others to test with.

2) Segmentation Pathways: Three different methods for segmentation were explored, including:

- a) Γ – traditional image segmentation;
- b) $\mathbf{MB}$ – multi-class U-Net;
- c) Σ – single-class U-Net;

Each of the above methods will be described below.

a) Γ : The traditional image segmentation algorithm developed seeks separately two semantic classes, namely the μ – marble class and β – bedding class. Since the marble pixels and bedding pixels exhibited vastly different textures, colors and sizes, two separate algorithms were applied to each segmentation task.

For the former, a block-based region-growing algorithm was applied. Seeding started within marbles, and neighboring pixels are iteratively incorporated into the region if pixel intensity compares within a threshold of the region mean intensity. For the latter, the perimeter of the bedding is sought. Several challenges exist with the bedding pixels, including varied textures, specular reflections, and image reflections on the cage walls. To that end, an edge-based segmentation approach was utilized. A Gaussian kernel was first applied to blur any rough textures. Canny edge detection was then applied. The result was normalized to binary representation, and the largest connected component was selected as the segmented bedding region. The above procedures are depicted in the first row of the Segmentation Pathways the Direct Prediction(s) column of Fig. 2. This method generates two direct predictions per input image, one to be evaluated against marble labels in μ , and the other against bedding labels in β .

b) $\mathbf{MB}$: The multi-class approach utilized two asynchronous yet identical U-net networks: marble segmentation network $\mathbf{M}$ and bedding segmentation network $\mathbf{B}$. Both networks take as inputs pre-processed images in γ , as depicted in Fig. 3, and are trained with different label classes.

In particular, $\mathbf{M}$ was trained with outputs from the μ – marble class, while $\mathbf{B}$ with those from the β – bedding class. These two networks are depicted in the second row of the Segmentation Pathways of Fig. 2. The features of a typical element from μ and β can be observed in the labels depicted in Fig. 3 and described in section II-A.2.b. This architecture is used to generate two direct predictions per input image, one to be evaluated against marble labels in μ , and the other against bedding labels in β .

c) Σ : The single-class architecture consists of a single U-Net. The inputs are preprocessed images from γ , as depicted in Fig. 3, and the model is trained with output labels from the σ – single-class label. This pathway is illustrated as the third row in Fig. 2, and is used to generate a single direct prediction per input image to be evaluated against single-class labels in σ .

3) Prediction Augmentation: Both pathways Γ and $\mathbf{MB}$ generate direct predictions of marble class μ and bedding class β . To create a σ class prediction from a pair of μ and β class, a region subtraction of the intersection of the two direct predictions from the bedding class is performed, the same method used to create single-class labels from marble and bedding labels. This is depicted in the right hand side of Fig. 3 and is described in section II-A.2.b.

In contrast, Σ generates direct predictions of class σ only. To generate a pair of μ and β class predictions, first an inverted marble label is computed. Consider the σ class shape as depicted in Fig. 3. To generate the marble label, the largest connected component is selected and inverted. This takes the white background and converts to black. Inverting the result creates a μ class prediction, and multiplying this with the original σ prediction yields the β prediction. All the above conversions and prediction augmentations were computed using Mathworks MATLAB image and binary image processing tools.

4) U-Net Training and Hyperparameters: The U-Net image segmentation networks were designed with the same hyperparameters and evaluated against the same labels. In particular, models were trained using the U-Net architecture with binary cross-entropy loss function, which is defined as

$$J = \frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (1)$$

where N is the size of the output (image pixels) and y and $\hat{y}$ correspond to binary true and predicted classes respectively. The training-testing split was heuristically determined as a 90-10 split. The total number of images was 526, and 10-fold cross-validation was implemented for evaluation without validation. a batch size of 3 for 10 epochs was used to train, with 300 steps per epoch. Early stop optimization was incorporated with a patience value of 3. Adam optimizer with learning rate of 1×10^{-4} was used. Augmentations such as rotation, horizontal and vertical shifting, zooming, horizontal flip and shearing were also implemented.

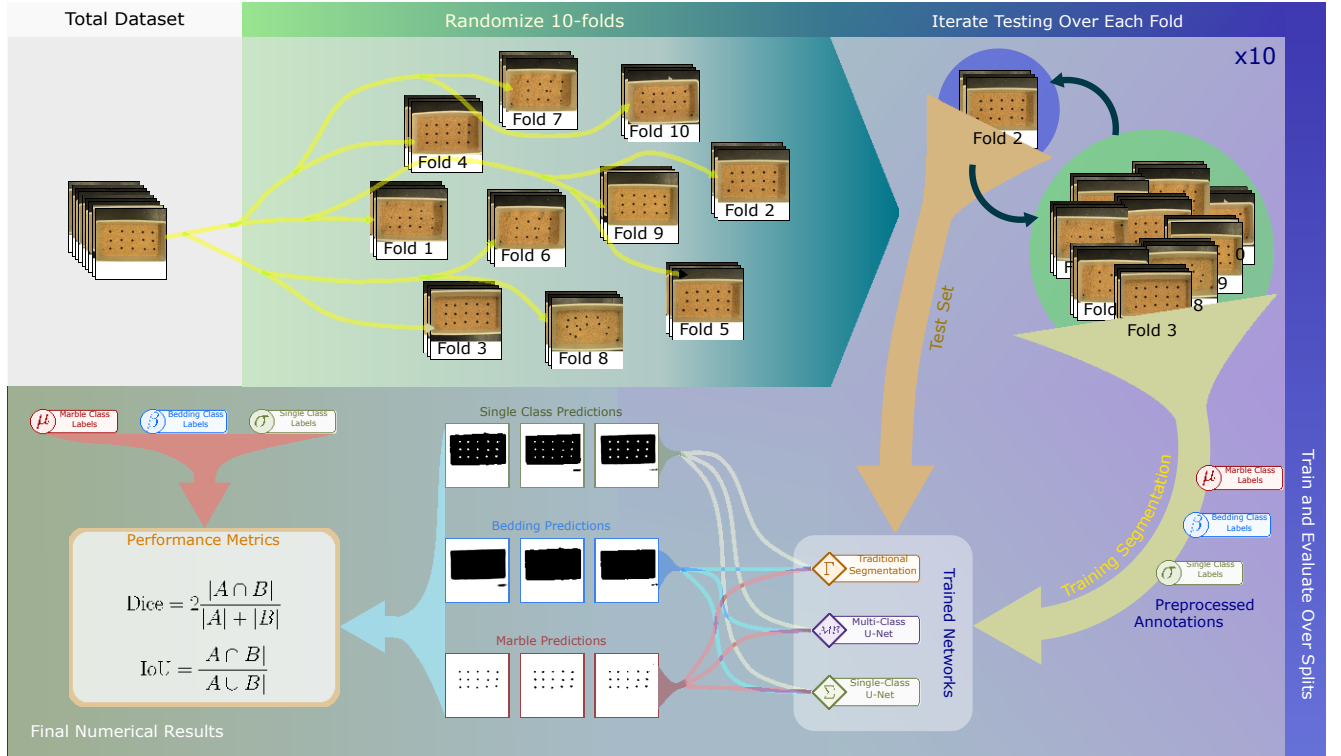


Fig. 4. Evaluation procedure: the dataset of 526 images is first randomly partitioned into ten roughly equal folds (52 elements each except the last which as 58). Each fold then iteratively acts as the test set, while the remaining folds are used as training data. The μB and Σ U-Nets are trained. Then, the test images in the concurrent fold are used as inputs to the trained networks as well as the traditional Γ approach. After prediction augmentation, each image in the test set generates 9 separate predictions, three for each method. These are then evaluated against the μ , β , and σ labels.

C. Segmentation Evaluation

Training as well as evaluation was implemented via 10-fold cross-validation with no validation step (the test fold was always for evaluation). Thus, each image in the dataset acts as a single test input exactly once. For each fold, the training set is used to train both the μB and Σ U-Net segmentation architectures. The test set images are then used to generate a prediction for each of the three classes for each of the three methods. These predictions are then evaluated against the class labels via Sørensen-Dice Coefficient and Intersection Over Union (IoU), which are defined as

$$\text{Dice} = 2 \frac{|A \cap B|}{|A| + |B|} \quad (2)$$

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

where A and B are prediction and label set respectively. The evaluation process is depicted in Fig. 4.

III. RESULTS

The mean segmentation performance results are summarized by testing metric, label type and segmentation pathway in Table I. Each of the 526 images serves as a test input exactly once. The resultant Dice and IoU calculations for each image were tracked across class and segmentation method. Figure 5 depicts each segmentation pathway's performance distribution across all images and label type. The same data is rearranged to show label type performance distribution amongst various segmentation pathways in Fig. 6.

TABLE I
SEGMENTATION EVALUATION

		Segmentation Pathway			
		Traditional Γ	Multi-Class M B	Single-Class Σ	
Metric	Dice	μ	0.757221166	0.88758079	0.928354095
		β	0.806576352	0.981457819	0.991926981
		σ	0.799630926	0.978751049	0.990742636
	IoU	μ	0.655794736	0.809805785	0.870823965
		β	0.682850268	0.966539918	0.984230333
		σ	0.673636875	0.96156552	0.981751322

TABLE II
MEAN DICE AND IOU SCORES

In total, 18 types of test result categories were computed - these are the entries in Table I. The correlation between prediction targets were sought within each metric type, i.e. Dice or IoU. Specifically, for each pair of label type (μ , β , σ) and segmentation pathway (Γ , μB , Σ), it was of interest whether other segmentation pathway scores for the same label target were correlated. For μ and β targets, it was also of interest whether the same segmentation pathway's own σ prediction was correlated with that prediction score. These results for both Dice and IoU are depicted in Fig. 7.

To visualize the segmentation performance across classes and segmentation pathways at the image index level, the Dice and IoU scores are displayed for each individual image in a heatmap shown in Fig. 8. This graphic provides an indication of any image that exhibits consistent poor segmentation across any segmentation pathway and target.

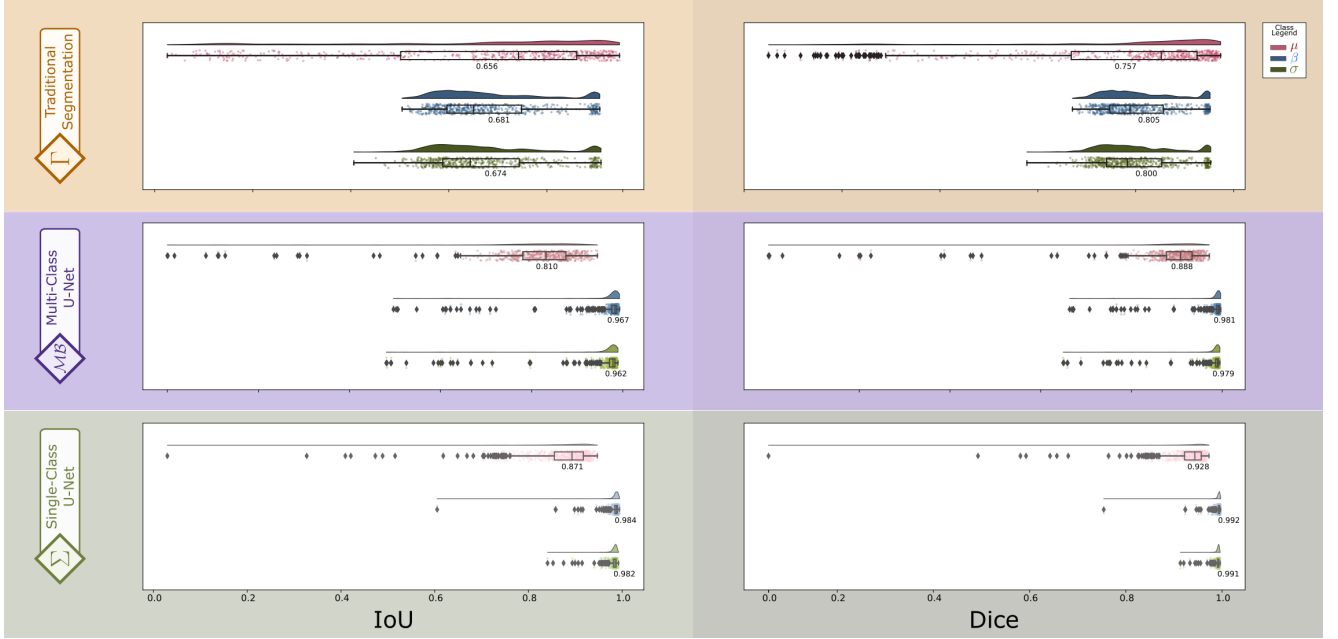


Fig. 5. IOU/Dice distributions organized by segmentation pathway. Shaded color areas denote segmentation pathways: Γ – traditional image segmentation; $\mathbf{MB}$ – multi-class U-Net; Σ – single-class U-Net.

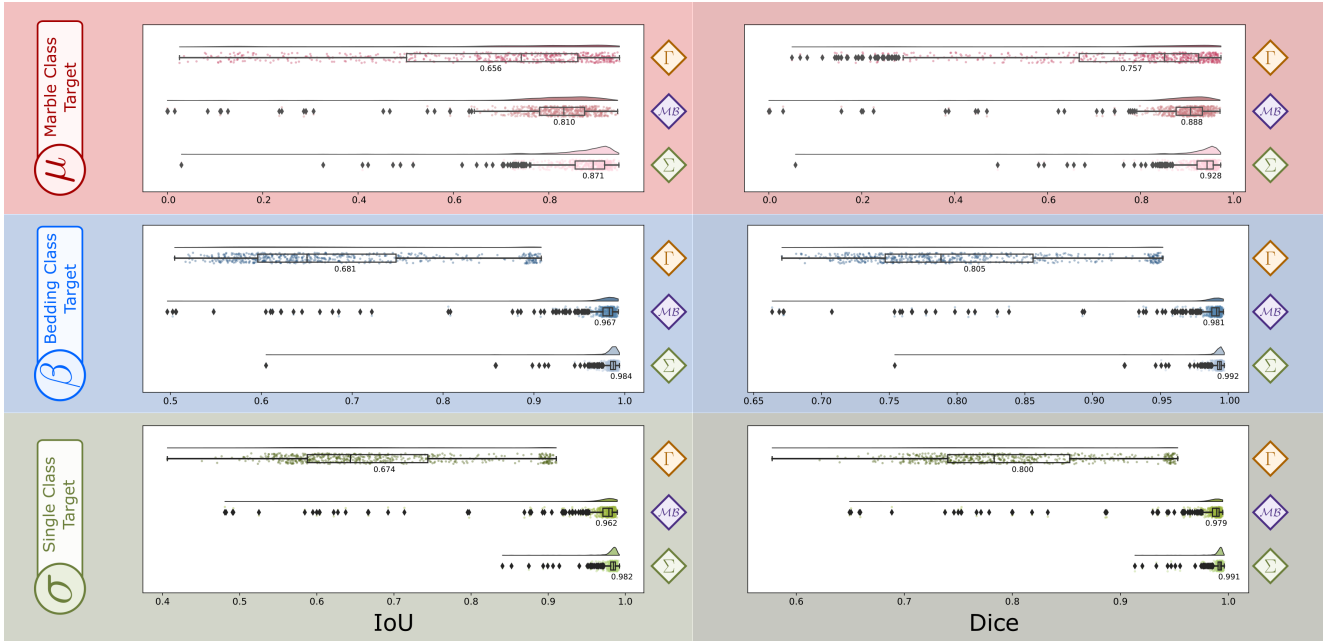


Fig. 6. IOU/Dice distributions organized by segmentation target. Shaded color areas denote segmentation targets: μ – marble class; β – bedding class; σ – single-class label.

A. Computational and Timing Considerations

For each U-Net ($\mathbf{M}$, $\mathbf{B}$, Σ), the training loss was plotted against training epochs for each of the ten folds, see Fig. 9. Training time was also tracked. The $\mathbf{MB}$ pathway consists of $\mathbf{M}$ and $\mathbf{B}$ as described in section II-B.2.b.

IV. DISCUSSION

The results in Table I show that the single-class network, Σ , generates across the board the best segmentation results (highest mean score), and the traditional, Γ , the worst. This, along with distribution of scores are observed in Figs. 5 and 6. Σ exhibits the least variance and thus best performance

consistency. On the other hand, Γ exhibits the least consistency. In all these observations, the multi-class architecture ($\mathbf{MB}$) performed similarly yet slightly poorer than Σ . From these figures, it is also noted that performance of β and σ predictions using traditional Γ techniques exhibit bimodal distributions. In contrast the deep-learning methods, $\mathbf{MB}$ and Σ , show unimodal scores. This is congruent with better generalizability of deep-learning methods over traditional ones. Among the three segmentation targets, the mean Dice and IoU scores and the spread of scores exhibit from better to worse (higher score, tighter spread), irrespective of segmentation pathway, the same order: β , σ , and then μ .

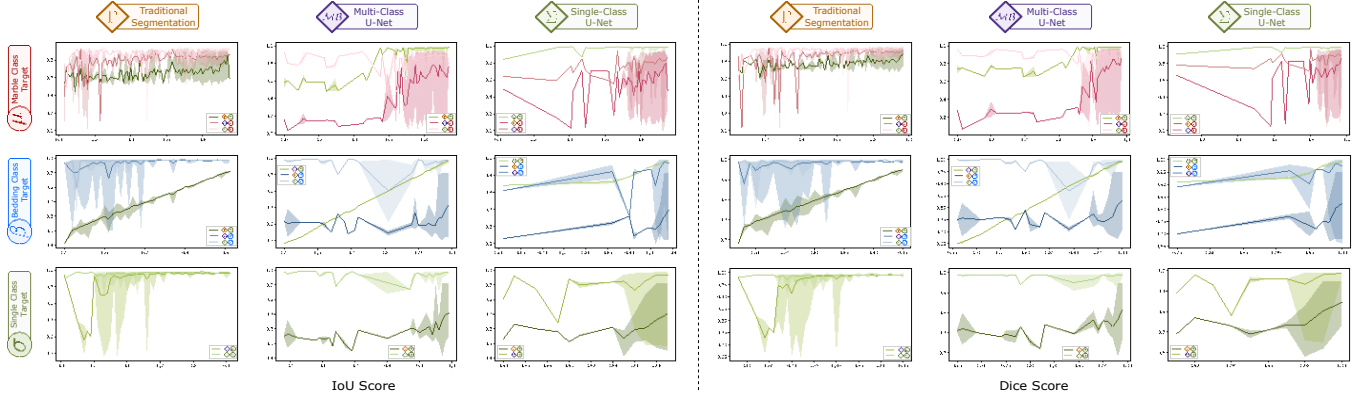


Fig. 7. Segmentation performance correlation within prediction target class (μ marble; β bedding; or σ single-class) using alternative segmentation pathways. In row 1 and 2, correlation with σ single-class target performance using identical segmentation pathway is also included. Tinted regions showcase the displayed alternative pathway metric score extremas given a score of the control segmentation method and target.

Figure 7 demonstrates the performance correlation within each prediction segmentation target class (μ , β , σ) using different segmentation pathways, along with correlation against the single-class, σ , prediction performance using the same segmentation pathway. Correlations are very similar between both examined metrics, i.e. Dice and IOU. In general greater positive correlation appears between the pair (β , σ) than (μ , σ) for any segmentation method. There is little observed correlation between (μ , σ) in the Γ and Σ methods.

Among the β predictions, there is consistently strong correlation between Σ and the two alternative pathways (row 2, column 3 of Fig.7). Meanwhile among the μ predictions, the only positive correlation is observed between the M B and Γ pathways (row 1, column 2 of Fig.7). Finally, higher scores (right) of the Γ subgraphs show a tight spread, whereas the Σ subgraphs show a wide spread. This suggests when the Γ pathway does well, the segmentation likely will perform robustly using alternative pathways. On the other hand, there is a greater performance variance using Γ or M B on images that performs well with Σ .

The segmentation performance scores for each image across all target classes and segmentation pathways were visualized in Fig.8. In general, an image is segmented least effectively using Γ compared to its counterparts using either M B or Σ . Within β , σ segmentation, pathways M B , Σ share an overall similar shade of blue across images. The main performance difference is the frequency of occurrences with poorly segmented outliers. Contrarily within μ segmentation, the Σ results show an overall darkest shade of blue, suggesting that Σ is generally the superior pathway for marble detection. Since Σ is the only pathway that trains with the joint label σ , it can be inferred that the bedding portion within label σ made marble identification easier.

Images that exhibit relatively poor metric scores (IOU or Dice) given a segmentation method and target can be seen as challenging images. These appear more consistent across target classes (μ , β , σ) given a segmentation pathway, than across segmentation pathways (Γ , M B , Σ) for a given target. While the challenging images of Σ appear to be a subset of that using M B , there is little observed correlation between challenging images of Γ with the two U-Net methods.

A. Timing Considerations

In terms of training time, Γ requires no training. It takes a total of 41851.5 seconds to train the two U-Nets, M and B , for the M B pathway, and only 13312.06 seconds to train the Σ model. Of the 41851.5 seconds to train the multi-class M B , only about 18% of that time was allocated for training the marble class portion, M . The U-Net training losses for models Σ , M , B are shown in Fig. 9. The marble segmentation network M (from M B) has the most consistent loss decrease across folds.

V. CONCLUSION

In this work, U-Net based deep-learning networks were trained to semantically segment marble-burying images to extract marble and bedding regions. Two U-Net methods were presented, a multi-class and a single-class method. These were trained with the same parameters, with promising performance as compared to a traditional method which comprised of a combination of region- and edge-based segmentation techniques. Furthermore, the single-class method outperformed the multi-class method in all segmentation quality metrics (IoU: 0.9818, Dice: 0.9907). Training the single-class network was also around 3 times faster than the

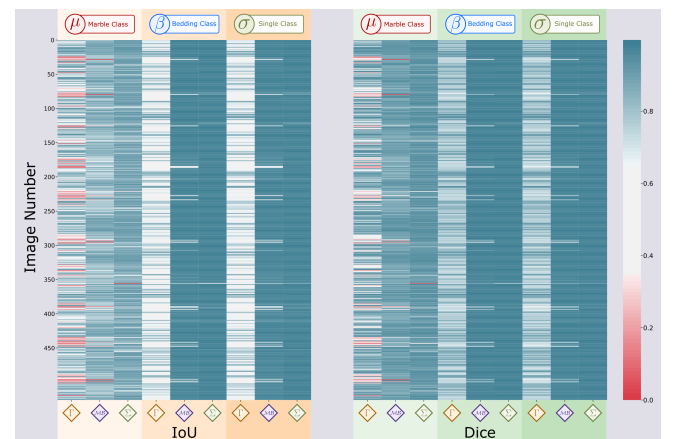


Fig. 8. Segmentation performance scores for each image in the dataset across all targets and segmentation pathways. Any single row that is generally lighter or has a red tone suggests an image that is not well segmented across any method or target.

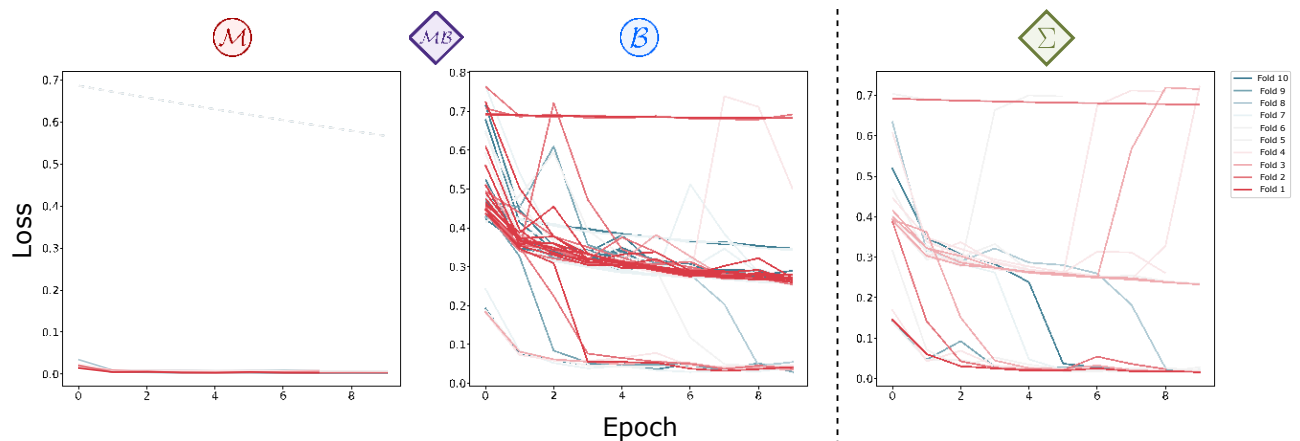


Fig. 9. Training loss plotted against epoch for each fold during training. Three total U-Nets were trained, with Σ and both M and B from pathway MB . An automatic model parameter reinitialization and training procedure was implemented for non-convergent training trials (dashed lines).

multi-class network. However, cost-benefit analysis should be considered given the use-case for marble-burying analysis. If, for example, only marble regions are of interest, training the sub-network for marbles in the multi-class architecture may provide sufficient performance at a relatively smaller cost of training time. Researchers should consider what semantic information from marble-burying is pertinent to their test before choosing a segmentation model. With successful marble and bedding region segmentation, future work may investigate novel bedding region features, such as bedding perimeter, as indicators of burying behavior.

REFERENCES

- [1] J. P. Pinel and D. Treit, "Burying as a defensive response in rats." *Journal of Comparative and Physiological Psychology*, vol. 92, no. 4, p. 708, 1978.
- [2] D. Sarkar et al., "A review of behavioral tests to evaluate different types of anxiety and anti-anxiety effects," *Clinical Psychopharmacology and Neuroscience*, vol. 18, no. 3, p. 341, 2020.
- [3] J. M. Witkin, "Animal models of obsessive-compulsive disorder," *Current protocols in neuroscience*, vol. 45, no. 1, pp. 9–30, 2008.
- [4] R. M. Deacon, "Digging and marble burying in mice: simple methods for in vivo identification of biological impacts," *Nature protocols*, vol. 1, no. 1, pp. 122–124, 2006.
- [5] G. T. Taylor, S. Lerch, and S. Chourbaji, "Marble burying as compulsive behaviors in male and female mice," *Acta Neurobiologiae Experimentalis*, vol. 77, no. 3, pp. 254–260, 2017.
- [6] H. Çalışkan, B. Şentunali, F. M. Özden, K. H. Cihan, M. Uzunkaulaoglu, O. Çakan, S. Kankal, and N. Zaloglu, "Marble burying test analysis in terms of biological and non-biological factors," *Journal of Applied Biological Sciences*, vol. 11, no. 1, pp. 54–57, 2017.
- [7] M. Angoa-Pérez, M. J. Kane, D. I. Briggs, D. M. Francescutti, and D. M. Kuhn, "Marble burying and nestlet shredding as tests of repetitive, compulsive-like behaviors in mice," *JoVE (Journal of Visualized Experiments)*, no. 82, p. e50978, 2013.
- [8] A. Thomas, A. Burant, N. Bui, D. Graham, L. A. Yuva-Paylor, and R. Paylor, "Marble burying reflects a repetitive and perseverative behavior more than novelty-induced anxiety," *Psychopharmacology*, vol. 204, no. 2, pp. 361–373, 2009.
- [9] C. Homma and K. Yamada, "Physical properties of bedding materials determine the marble burying behavior of mice (c57bl/6j)," *The Open Behavioral Science Journal*, vol. 3, no. 1, 2009.
- [10] N. Cheng, S. A. Masino, and J. M. Rho, "Ketogenic diet, social behavior, and autism," *Ketogenic Diet and Metabolic Therapies: Expanded Roles in Health and Disease*, p. 154, 2022.
- [11] D. N. Ruskin, M. I. Murphy, S. L. Slade, and S. A. Masino, "Ketogenic diet improves behaviors in a maternal immune activation model of autism spectrum disorder," *PloS one*, vol. 12, no. 2, p. e0171643, 2017.
- [12] D. N. Ruskin, I. C. Sturdevant, L. S. Wyss, and S. A. Masino, "Ketogenic diet effects on inflammatory allodynia and ongoing pain in rodents," *Scientific Reports*, vol. 11, no. 1, pp. 1–8, 2021.
- [13] L. Wahl, A. M. Punt, T. Arbab, I. Willuhn, Y. Elgersma, and A. Badura, "A novel automated approach for improving standardization of the marble burying test enables quantification of burying bouts and activity characteristics," *Eneuro*, vol. 9, no. 2, 2022.
- [14] N. M. Zaitoun and M. J. Aqel, "Survey on image segmentation techniques," *Procedia Computer Science*, vol. 65, pp. 797–806, 2015.
- [15] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Ernet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.
- [16] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5229–5238.
- [17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [18] K. Huang, D. Chitrakar, W. Jiang, and Y.-H. Su, "Enhanced u-net tool segmentation using hybrid coordinate representations of endoscopic images," in *2021 International Symposium on Medical Robotics (ISMR)*. IEEE, 2021, pp. 1–7.
- [19] K. Huang, D. Chitrakar, W. Jiang, I. Yung, and Y.-H. Su, "Surgical tool segmentation with pose-informed morphological polar transform of endoscopic images," *Journal of Medical Robotics Research*, 2022.
- [20] Y.-H. Su, W. Jiang, D. Chitrakar, K. Huang, H. Peng, and B. Hannaford, "Local style preservation in improved gan-driven synthetic image generation for endoscopic tool segmentation," *Sensors*, vol. 21, no. 15, p. 5163, 2021.
- [21] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.
- [22] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, "H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes," *IEEE transactions on medical imaging*, vol. 37, no. 12, pp. 2663–2674, 2018.
- [23] Y.-H. Su, I. Huang, K. Huang, and B. Hannaford, "Comparison of 3d surgical tool segmentation procedures with robot kinematics prior," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4411–4418.
- [24] M. Allan, A. Shvets, T. Kurmann, Z. Zhang, R. Duggal, Y.-H. Su, N. Rieke, I. Laina, N. Kalavakonda, S. Bodenstedt, et al., "2017 robotic instrument segmentation challenge," *arXiv preprint arXiv:1902.06426*, 2019.
- [25] "marble_unet," 8 2022. [Online]. Available: <https://github.com/TrinityPandaLab/marble-unet>