Learning Progression-based Automated Scoring of Visual Models

Ari Sagherian*, Suhasini Kalaiah Lingaiah*, Mohamed Abouelenien*, Chee Wee Leong¶, Lei Liu¶, Mengxuan Zhao¶, Blake Lafuente*, Shu-Kang Chen¶, Yi Qi¶
*Computer and Information Science, University of Michigan-Dearborn
Dearborn, Michigan, USA
¶Research & Development, Educational Testing Service (ETS)
Princeton, New Jersey, USA
<asagheri,suhasikl,zmohamed,balaf>@umich.edu
<cleong,lliu001,mzhao,tschen,yqi>@ets.org

ABSTRACT

Visual models are defined as drawings created by students to illustrate their understanding of an observed scientific phenomenon. Along with the corresponding textual answers describing the phenomenon, these multimodal responses serve as rich vehicles of information for conducting assessment on students in their grasp of the underlying scientific concept. Though effective, manual scoring of these responses are both laborious and expensive. In our work, we apply a user interface tool for students to construct multimodal responses to scientific prompts, and propose an automated approach that relies on image processing to classify shapes in visual models, extract relevant features, and, ultimately, assign a learning progression score to each model. This effort is the first in a series of planned approaches targeted at utilizing multimodal evidence to evaluate, scaffold and guide students in their learning pathways in science education.

ACM Reference Format:
Ari Sagherian*, Suhasini Kalaiah Lingaiah*, Mohamed Abouelenien*, Chee Wee Leong¶, Lei Liu¶, Mengxuan Zhao¶, Blake Lafuente*, Shu-Kang Chen¶, Yi Qi¶. 2022. Learning Progression-based Automated Scoring of Visual Models. In *The15th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '22), June 29-July 1, 2022, Corfu, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3529190.3529192

1 INTRODUCTION

In the United States, there has been an increased emphasis on Science, Technology, Engineering, and Mathematics (STEM) education, of which an important subtask is the assessment of students' mastery of science competencies. Assessment experts have been researching on innovative measures that could potentially evaluate the multiple dimensions of science competencies, such as scientific concepts and practices [10]. One assessment approach, Learning progressions (LPs), was developed to facilitate the assessment of students' progress in science learning [8, 21]. These LPs facilitate the aggregation of scores of each progress variables, where each focuses on a specific facet of the overarching concept or practice.

Relatedly, the Next Generation Science Standards (NGSS) designated constructing models as an essential practice as learning targets for K-12 science learning [9]. By definition, visual models are drawings created to illustrate understanding of an observed scientific phenomenon or mechanisms that explains a phenomenon, e.g., ocean water modeling task in Figure 2. These visual models help capture alternative evidence about students' understanding of science in combination with their corresponding textual responses in a multimodal response. LPs were developed and could be used to interpret the meaning of student-generated visual models and to evaluate their understanding of the structure and properties of Matter [17, 29]. In addition, these visual models also introduce other elements characterized by artistic creativity and complexity. Consequently, the overarching motivation behind this work is to disentangle the interaction between the scientific modeling skills and artistic skills of representing real objects to provide a fair and valid way to assess understanding of scientific concepts.

Automated scoring of these visual models and their accompanying textual answers is a worthy pursuit since manual scoring is both laborious and expensive. In this paper, we seek to achieve the two primary goals of (1) creating accurate classifiers of objects and extracting relevant features from visual models rendered as unlabeled image, and, (2) conducting assessment of the level of mastery of students of a targeted, scientific concept, i.e., Matter, using the features of the classified objects. The paper is organized as follows: We first introduced important datasets used in our studies and highlight their characteristics. Next, we performed a comparative evaluation between supervised and unsupervised approaches used to extract features from the visual models and classify objects. Finally, an additional level of features was aggregated from the previous layer of feature extraction and shape classification, and used to assign a learning progression score to each visual model. Note that this paper scopes a first-cut, "depth-first" approach to solving a multimodal problem first using image processing prior to its combination with natural language processing technique at a later point. To our knowledge, this represents the first fully-automated pipeline for assessment of visual models in an end-to-end manner.

2 RELATED WORK

The conceptual foundation for automated scoring in the literature was first established by Liu et al. in their research on grading

PETRA '22, June 29-July 1, 2022, Corfu, Greece
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9631-8/22/06…$15.00
https://doi.org/10.1145/3529190.3529192

scientific visual models [24]. They administered questions to students who, in response, created visual models to illustrate their understanding of the targeted scientific concept, e.g., Matter and related phenomenon. These models were then evaluated by four progress variables as part of the Learning Progression (LP) framework, with levels ranging from 1 (lowest) to 5 (highest) on a Likert scale [8]. Each visual model consisted of multiple 2-D shapes arranged spatially in the drawing to illustrate understanding of the targeted scientific concept phenomenon, e.g., water particles in ocean water. Each of the four progress variables, namely Scale, Behavior, Material Identity, and Distribution, had their own subprogression of 5 levels, with the amalgam of these resulting in the final score ("LP" score).

In order to automate the scoring of visual models, object classification and feature extraction techniques are necessary. Various strands of work in image processing provide a wide array of candidate techniques that can be used. Many begin with pre-processing steps to improve results. Image segmentation is one such example which can be divided into four categories, as described in [14], with thresholding techniques being the most relevant to this study. Threshold segmentation is based on the idea that pixels in an image within a certain range of color values are part of the same class [22, 31], thus improving the signal to noise ratio between objects.

After preparing the data, many studies utilized contour or edge detection as the next step in the pipeline of object recognition. The most commonly used include the Suzuki [40] and Seo [38] contour detectors or the Sobel [39], Roberts [35], Prewitt [34], and Canny [7] edge detectors. Each of these possesses its own benefits when applied to a variety of object recognition tasks as described in [1–3, 12, 23, 27, 28, 30, 32]. A prominent example of a classification tool following the usage of contour detection is Hu Moments. These are seven values that describe an object in a scale, rotation, and translation invariant manner [16, 18]. These values can then be used as features to compare and recognize objects [15, 41, 42].

An additional host of tools after isolating the objects are supervised machine learning approaches. For example, common feature extractors for object recognition such as the Scale Invariant Feature Transform (SIFT), Speeded up Robust Feature (SURF), and the Oriented FAST and Rotated BRIEF (ORB) algorithms [4, 25, 36] can be implemented. Then, these extracted features can be fed into a supervised classifier, such as the k-Nearest Neighbors (KNN) and Support Vector Machine (SVM) algorithms [5, 6, 11]. When comparing the three feature extractors with a KNN classifier, it was concluded in [19] that SIFT provided the best overall accuracy but took significantly longer than ORB, leaving the SURF approach with a moderate performance. Therefore, if speed is essential, then ORB is the ideal candidate and, if not, then SIFT is the ideal choice. Comparing classifiers in the context of object classification, [20] determined that SVM's tend to outperform the KNN classifier. The advent of machine learning relies on large datasets but, once possessed, has unlocked the ability to improve the accuracy of classification and holds the potential for future improvement.

3 ASSESSMENT PROTOTYPE

The assessment prototype was developed to elicit sufficient evidence to locate student response along learning pathways defined in the LP. Specifically, this assessment prototype was designed to measure the LP levels 1-4. As in most LPs, the higher anchor is often the ultimate learning goal, and this rationale will be covered later in the next section. The assessment prototype is a scenario-based task beginning with a driving question—"How can you get pure water out of ocean water?"—to set the ultimate goal for the whole task. In addition, the assessment prototype included tools and representations similar to scientists tend to use, including simulations and a modeling tool. Throughout the task, students engaged in interactive simulations that allow students to design multiple trials of experiments to test predictions and explain why they think these experiments can test the provided prediction. Item formats include multiple choice, constructed response, and modeling items.

For the purpose of this paper, our analyses focused on student drawings from the modeling items. In this assessment prototype, modeling items involve the use of a computer-based drawing tool in Figure 1, in which students used a free drawing tool or select from a pool of predefined objects, including abstract representations (e.g., circle, square, or triangle) and concrete representations that include common misconceptions that students hold (e.g., fish and water drops are the basic units of ocean water composition) to allow students to express their idea of structure of matter. The common misconception objects were collected from a cognitive lab study. The drawing tool also allows students to change the size or color of selected objects, add arrows to represent motion, and label objects. Finally, the drawing tools include two modes of inputs, namely images and text, as research shows that it is necessary to ask students to provide textual description of their models to avoid potential misinterpretation of student drawings [37].

The modeling tool was embedded at several places in the task so that students can construct, use, evaluate and revise their models. At the beginning of the task, students were informed of the importance of modeling in learning about science and provided with an example of modeling to demonstrate how scientists used models to refine theories. Then, students were asked to use the modeling tool to draw a model of pure water particle model and ocean water particle model. Then they were asked to evaluate one classmate's model based on the evaluation criteria provided. Three simulation-based activities (i.e. filtering activity, evaporation activity, and condensation activity) were designed to break down the big problem into smaller ones and to guide students to reach the solution of the driving question—"How can you get pure water out of ocean water?" At the end of each activity, students were asked to reflect on the activity and use the modeling tool to draw a model at the particle level to explain what happened (e.g., why pure water and ocean water had different densities, how water evaporate and condense?). In addition, students were also prompted to revisit their initial models of pure water and ocean water and revise them if needed. Finally, at the end of the task, the students were asked to draw a model of their solution of purifying ocean water.

4 DATASET

Three datasets are used in this paper and comprises multiple scientific visual models, provided by the ETS organization. The first is the **Ocean Water Modeling Item** dataset that consists of

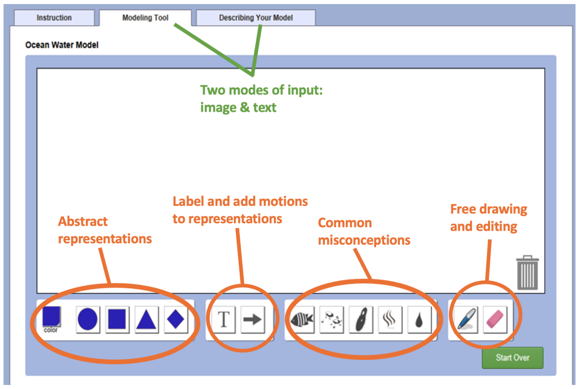


Figure 1: User Interface screenshot for a sample visual modeling task

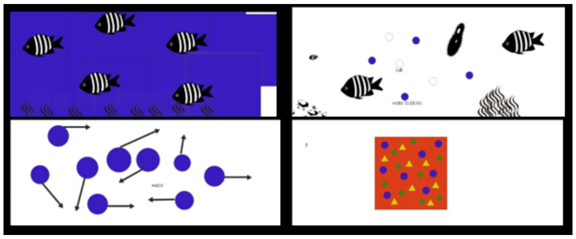


Figure 2: Examples from the Ocean Water dataset.

144 visual models. In this dataset, students illustrate their understanding of composition particles in ocean water through drawing the smallest unit of matter included in a water droplet by spatial placement of provided micro-objects (triangles, squares, rectangles, circles, arrows) or macro-objects (fish, seaweed, water droplets, etc.) on a digital canvas. According to a previously developed LP, a low level understanding tends to include more macro-objects, while more sophisticated understanding is exhibited through drawings of micro-objects and associated behaviors of those objects at the particle level. Students were instructed to select any number and type of micro-objects and macro-objects to create a visual model to illustrate their understanding of the targeted phenomenon. This dataset has human annotations for each of the individual shapes comprising the models, as well as the final “LP” score for each model. Human annotations are used as the ground-truth labels for comparing both supervised and unsupervised approaches to object detection. The annotations ranged from LP score 1-4. Examples of visual models from the Ocean Water dataset can be seen in Figure 2.

The second dataset, **Synthesized Ocean Water**, consists of 30 visual models that are artificially synthesized and correspond to a mastery of the targeted scientific concept at the highest understanding level i.e. LP 4, as measured by the four progressive variables, i.e., Scale, Behavior, Material Identity, and Distribution. For a given visual model, the scale dimension measures understanding of composition of Matter beginning with the smallest units, e.g., nanoscopic particles. The material identity dimension examines the anticipated number/identity of particles present. The behavior dimension examines if/how particle movement is represented. Finally, the distribution dimension examines positions of individual particles and space between them in a given Matter state. Each dimension has its own sub-progression levels, starting with the most basic at level 1. Overall, in order to reach a given level of the LP, the student must demonstrate a minimum level of thinking in each of the four dimensions, and that minimum level may vary with the dimension. For instance, the progression from LP-3 to LP-4 requires a mastery of level 4 in scale dimension and a minimum of level 3 in behavior dimension, as illustrated in Table 1. Only annotations for LP scores, not individual shapes within each visual model, are available for this dataset.

Table 1: Mapping between Learning Progression (LP) levels to four dimensional sub-progressions. For a given LP level, indication of a ‘X’ means the minimum level that must be mastered in that sub-progression dimension. For example, a student with a visual model worthy of a LP-4 score must exhibit understanding that commensurate with at least the following sub-progression level scores: Scale (4), Material Identity (2), Behavior (3), Distribution (2)

	S				MI			B				D		
	1	2	3	4	1	2	3	1	2	3	4	1	2	3
LP-1	X	X			X	X		X				X		
LP-2		X	X			X		X	X			X		
LP-3			X	X		X	X		X	X			X	
LP-4				X		X	X			X	X		X	X

Finally, the **Two Can** dataset consists of 195 visual models ranging from LP levels 1-3, and has annotations for LP score of the models only, similar to the Synthesized Ocean Water dataset. Here, students illustrate their understanding of the water condensation phenomenon through the use of micro-objects and free-hand drawing of two cans, one warm and one cold. Students with a deep understanding of the phenomenon are expected to draw a visual model with condensation effect on the surface of the cold can and not the warm can. Furthermore, the Two Can dataset contains hand-drawn objects and words, which poses additional challenge for image processing. Examples of the Synthesized Ocean Water and the Two Can datasets are shown in Figures 3 and 4 respectively.

With both shape and LP score annotations, the 144-image Ocean Water dataset can therefore be used to evaluate our shape classification algorithms and frameworks. However, it lacks visual models annotated at the LP-4 level. Hence, it can be combined with the Synthesized Ocean Water dataset to form a 174-image dataset that can be used to evaluate our ability to predict the LP score of each visual model in a cross-validation setting. The Ocean Water and Two Can datasets can also be used to evaluate our ability to predict the LP score of a visual model in an *out-of-domain* setting, since these drawings represent different phenomenon, i.e., water and salt particle movement in a liquid state versus water particle movement in a gaseous/condensation state.

As previously noted, except for the Synthesized Ocean Water dataset, there is an supplemental textual answer provided by the student accompanying each visual model, with varying degrees in the text length and focus of the description, e.g., *“Our model shows*

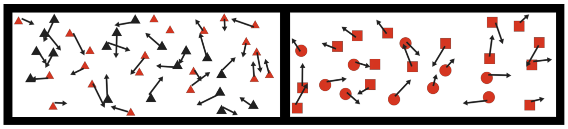


Figure 3: Examples of visual models from the Synthesized Ocean Water dataset.

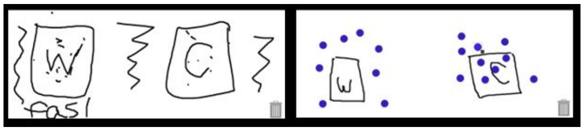


Figure 4: Examples from the Two Can dataset.

the smallest unit of water, under a microscope. The arrows show the way the water is going.". This textual modality presents additional evidence for assessment of understanding by students but exceeds the scope of this paper, and would be addressed in ensuing work.

5 METHODOLOGY

The goal of this study is to accurately recognize various shapes in 2D visual models and extrapolate LP scores from the classified objects and their features. In order to achieve this goal, we next describe the processing pipeline of our methodology. Our approach can be divided into two main stages. The first involves processing the raw visual models using a series of image processing and segmentation techniques, as well as unsupervised and supervised approaches to reliably detect the drawn objects in each model. The second stage is concerned with aggregating the results of the first stage into higher level features that model the mastery of the targeted scientific concept through visual drawings via a Learning Progression (LP) score.

5.1 Image Preprocessing

To begin the pipeline of processing a visual model, image processing techniques are implemented. One such technique is color segmentation which refines the colors so that the contrast between segments is improved. The color segmentation is applied identically to all objects in the visual models. The first step in segmenting the images for this study was to count the number of pixels with the same color. A filter was then applied to remove all colors that appeared less than 200 times per image in order to reduce noise. Having filtered out low frequency colors, the next step was to join the pixels that were within values of 40 in intensity to specify a limited list of colors and increase the contrast. This number was determined experimentally. Having segmented the images by color, the final steps of image preprocessing were converting the image to grayscale, followed by inverse-binarization of the image.

5.2 Feature Extraction via Contour Detection and Shape Approximation

The next step in the pipeline was to further process the images with contour detection to extract features. The Suzuki algorithm was implemented, which took a binarized image as an input and then listed all the closed-loop contours. Then the approximate polygon of each closed contour was determined using the Ramer-Douglas-Peucker (RDP) [13] algorithm. The benefit of this approximation is that the number of vertices in a contour can be determined, a useful feature which will be exploited in the following sections.

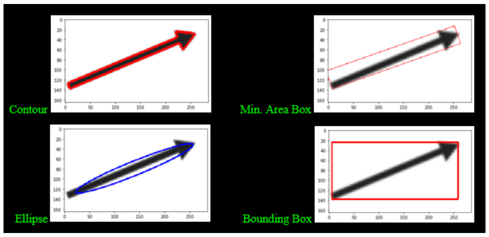


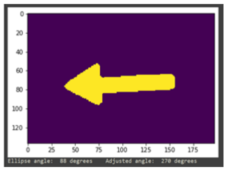
Figure 5: Contour and bounding shape examples for arrows.

We illustrate the process described above through a walk-through example focusing on a specific object, i.e. arrow. In this case, two bounding boxes and an enclosing ellipse were automatically created around the contour (top left), as shown in Figure 5. One bounding box (bottom right) has sides parallel to the frame of the image, providing the center coordinates of the object which was used for the final orientation calculation. Another box (top right) is a minimum area bounding rectangle which encapsulated the contour with the smallest possible rectangle, providing the height, width, coordinates, and rotation of the object in a range of 0 to 90 degrees relative to the horizontal. The bounding ellipse (bottom left) provides further rotational information, calculating the angle from 0-180 degrees relative to the vertical.

5.3 Unsupervised Methods

5.3.1 Rules-based Shape Classification. To classify the detected objects as different shapes, we experimented with several approaches. The first fully unsupervised approaches (i.e. without any training inputs) is a Rules-based classifier that we designed, which used the number of vertices returned by the approximate polygon and the height, width, and rotation returned from the bounding boxes. The rest of this section lists the shape, and the rules we used to classify them: **Triangle**: the number of vertices equals 3 or 5, the height cannot be equal to the width, and the height is within 10 pixels of the width. These values, including 5 vertices, although not typical of triangles, were experimentally determined due to the pre-processing steps. **Diamond**: the number of vertices equals 4, the rotation of the bounding box is between 40 and 50 degrees, and the height of the box is within 2 pixels of the width. **Square**: the number of vertices equals 4, the shape is not a diamond, and the height is within 2 pixels of the width. **Circle**: the number of vertices is greater than 6 and the height is within 1 pixel of the width. **Arrow**: the number of vertices is between 7 and 10, color is black (for our datasets), and the height is not equal to the width. **Other**: other objects that do not satisfy the above rules.

5.3.2 Arrow Orientation. Another novel contribution of this work is a new arrows orientation detector. Students were instructed to use arrows to indicate water particles motion in terms of both their direction and speed. Hence, a visual model with appropriately



```
Ellipse angle:  88 degrees    Adjusted angle:  270 degrees
```

Figure 6: Adjusted arrow orientation using 3-degree margin of error

drawn arrow directions and length would represent a deep understanding of the targeted scientific phenomenon at the particulate level compared to a visual model with macroscopic depiction (e.g. fishes with seaweed). Hence, the ability to detect the direction of the arrows drawn, as well as the length of the arrows (which corresponds to the speed of the particle to which the arrow is attached) was an important feature engineering step. To our knowledge, an effective method for determining arrows orientation in the context of this study does not exist. The first step in our approach for calculating the orientation was to take the contour of each candidate arrow in an image and fit a Rectangular Bounding Box, a Minimum Area Bounding Box, and a Minimum Area Ellipse around it. Each of these had center coordinates that can slightly vary based on the angle of the arrow, so the average of all three was calculated and used as the geometric center of the arrow. Then, the centroid of the arrow was derived to determine the coordinates of the arrow's center of mass.

Equipped with the geometric center and the centroid, the next step was to use both of these to determine the direction of the arrow's head. The principle guiding this approach is that the centroid (relative to the geometric center) was shifted towards the arrow's head, analogous to a see-saw tilting in the heavier direction. Using this insight, another Rules-based approach was created to decide the orientation of each arrow.

Initially, the angle of each arrow was calculated between 0-180 degrees relative to the vertical axis derived from the Minimum Area Ellipse. For consistency, this angle will be called the ellipse angle for the remainder of this section. In this scenario, an arrow facing North and an arrow facing South were both rotated zero degrees. Three sets of rules were created to adjust for this, which determined the angle for arrows in 90 degree intervals facing North, South, East, and West based on a three-degree margin of error from the ellipse angle. An example of the implementation of these rules can be seen in Figure 6. We applied two other sets of rules to account for instances, where the centroid and geometric center were within a range of one pixel and when the centroid was west of the geometric center. The arrows orientation was one of the important attributes used to build the second-level features that were used to determine the learning progression score, as detailed later.

5.4 Supervised Methods

5.4.1 Machine Learning. In order to evaluate the performance of a supervised approach, we extracted features which are fed into a shape classifier. Since the amount of labeled, ground-truth data is limited for this project, Decision Tree is the only classifier that is currently explored due to its established statistical properties. We intend to use this as a basis for future work on classifying shapes using other supervised learners as more labeled data is collected.

The Decision Tree classifier operates by taking feature vectors extracted from the visual models along with the corresponding classification labels as the training inputs. It then receives the testing feature vectors, which have identical attributes but with different values. The features from the training inputs are then ranked in terms of their information gain, and used to predict the classification labels in the test set. For purposes of this paper, the final classifications will be the object type, which are extracted from the same templates used for Hu Moments and Area Ratios, are as follows:

(1) Number of edges in the contour's approximate polygon
(2) Ratio of bounding box area to contour area
(3) Ratio of minimum area bounding box to contour area
(4) Ratio of bounding ellipse area to contour area
(5) Minimum area bounding box rotation
(6) Bounding ellipse rotation
(7) Distance between the geometric center and the centroid

Consequently, these features were used to create a single feature vector for each shape in each visual model, which was used along with its correct label to train the Decision Tree classifier in a 5-fold cross validation scheme. The metric used to evaluate the results is the classification accuracy. The accuracy of this approach accounted also for shapes that missed detection in the first step.

5.4.2 Hu Moment Classification. While Hu monents and the following Area Ratio approaches, by themselves, might not be considered fully supervised, they are listed under supervised approaches, as we added a step where pre-selected templates were used to determine the type of the target shape in the images. However, very few templates were used for these approaches. Hu Moments are seven values that describe a shape in a scale, rotation, translation, and reflection invariant manner.

By undergoing various linear algebraic transformations, these invariant properties were achieved. The Hu moments were calculated for the outermost contour in a template image (as each template represented a single shape) and for each contour in the visual models. To determine which shape the detected contour in the visual model belonged to, distance measures were used between the Hu moments of the visual model contour and all the template images contours.

5.4.3 Area Ratio Classification. The implementation of three Area Ratios is another approach we designed to classify the detected objects. The ones used in this project are the ratios between the detected contour's area and the area of three enclosing shapes, namely the Bounding Box, Minimum Area Bounding Box, and Minimum Area Ellipse, as shown earlier in Figure 5.

Dividing the areas of these three bounding shapes by the target contour area generated three Area Ratios. This process was repeated also for each template, resulting in target and template contour Area Ratios. The respective ratios of the templates and detected contours were then compared using the distance operator shown in Equation 1. This distance measure represented the absolute value of the difference between the template Area Ratio (p) and the shape's contour Area Ratio (q). Similar to Hu Moments, the

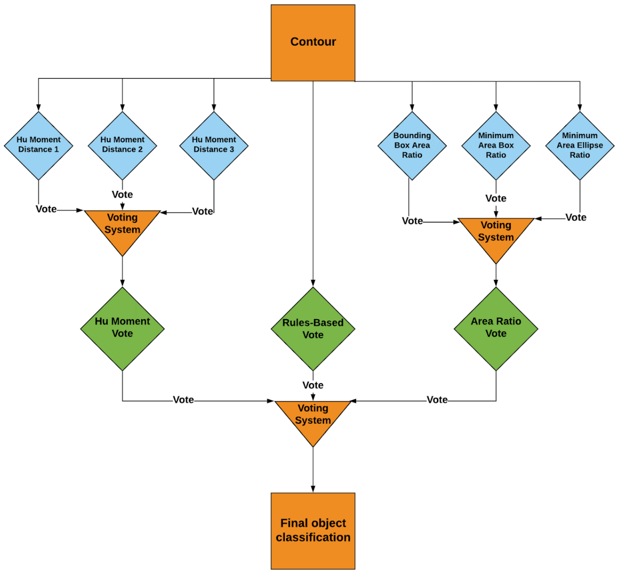


Figure 7: Flow Chart of the cascaded voting system

smaller the difference is, the higher the probability of a match. As the distance measure for each of the three area ratios was conducted independently, it was possible for all three to produce differing results. In this case, the Minimum Area Bounding ratio was used as it was experimentally determined to be the most robust.

$$D = |p - q| \tag{1}$$

5.5 Cascaded Voting System

To reach a reliable classification of a shape, we designed a novel cascaded voting system using four main stages to integrate three of the aforementioned methods, namely, Area Ratios, Hu Moments, and Rules-based methods. These methods were selected as they are either completely unsupervised or require very few templates or labeled shapes.

Moreover, they can be easily expanded to use with any dataset. Three of the four stages resulted in a vote for each of the Area Ratios, Hu Moments, and Rules-based methods, and the last stage integrated their vote to result in a final shape classification.

At the first stage, each of the three Area Ratios for the target object was matched with the most similar template. The final vote of each Area Ratio was determined using a majority voting system for the top 1, 3, and 5 results. For example, the list of matched templates could be [Square, Circle, Circle, Square, Square]. For the top 1 voting schema, the result will be Square since the most similar match is a Square. For the top 3 schema, the result will be Circle as 2/3 of the top results are Circles. For the top 5 schema, the result will be Square, following the same approach. We used all three schemata and employed an additional layer of majority voting among the three for a more reliable vote for each Area Ratio. For the previous example, the decision using majority voting on all three voting schemata for a given Area Ratio should be a Square. Lastly, the final vote of all three Area Ratios was also determined using a majority voting scheme of their three individual votes.

Similarly, the three votes obtained using three distance measures with Hu moments were integrated with a majority voting scheme, resulting in one final Hu Moment vote. The next level of the cascade integrated the Hu Moment and Area Ratio votes with the Rules-based vote using majority voting once again. For instance, if the Rules-based and the Hu Moment classifications match, then the final classification of the object is whichever shape these two methods predicted. A diagram of the cascaded voting system is illustrated in Figure 7.

5.6 Scoring of Visual Models

The second stage of the pipeline is concerned with the assessment of the visual models. The features and classifications from the prior stage were aggregated to a second level of 10 features that are aligned with the constructs of mastery of the targeted scientific concept, as shown in Table 2.

Table 2: Descriptions of features for scoring visual models

Features	Description
	Counting-based
Micro-object types	Number of each type of micro-object
Macro-object types	Number of each type of macro-object
Micro-object color types	Number of types of micro-object colors
EIC deviation	EIC minus count of each type of micro-object
Arrows	Count of arrows
Arrow lengths	Mean length of arrows
Arrow randomness	Variance of arrow orientations
	Spatial-based
Spatial-based k-NN = 3	Mean distance to 3 nearest micro-objects
k-NN = 10	Mean distance to 10 nearest micro-objects
Dispersion	Mean normalized spread of micro-objects

The final step in the second stage is using these aggregated features to predict the Learning Progression (LP) score. For this study, the LP scores ranged from 1 (lowest) to 4 (highest). The LP used for this study consisted of four progress variables, each with their own sub-progressions as displayed in Table 1, where the combination of these sub-progression scores mapped to the overall LP score. The progress variables are Scale (S), Material Identity (MI), Behavior (B), and Distribution (D). The Scale portrays the composition of Matter based on the micro and macro objects. The Material Identity portrays the number and identity of the objects used in the visual model. The Behavior dimension measures the movement of the objects relative to each other. Finally, the Distribution dimension examines the spatial locations and clustering of objects.

The score prediction process utilized eight regressors and classifiers known for their matured statistical properties and explainable outputs, which are recommended for educational assessment tasks, as shown in the list below. We employed these various learners in the SciKit-Learn machine learning framework ([33]) via SKLL ([26]) for experimentation. The results of these learners were tuned to a Quadratic Weighted Kappa (QWK) score which compares the predicted results to human-annotated LP scores. The range of QWK scores is from 0 to 1, where 0 indicates a completely random prediction and a score of 1 represents a complete agreement between predicted and ground-truth labels.

5.7 Generalizability and Conceptual Understanding

The classification of shapes and extraction of their corresponding features with this system are not limited to the Ocean Water and Two Can datasets. Due to the broad use of micro-objects including arrows, squares, etc., the current system can also be generalized to other datasets for predicting LP scores. The generalizability of this system for other objects further allows it to automatically score a variety of visual models and effectively measure students' understanding of scientific concepts in a fast, scalable, and robust manner.

6 EXPERIMENTAL RESULTS

To evaluate our approaches, the experiments can be split into two phases. The first phase is the evaluation of correct shape classification within the visual models. The second phase is the prediction of Learning Progression (LP) scores and their agreement with the human score annotations.

6.1 Shape Classification

The first part of this work evaluates multiple approaches to classify shapes and extract proper features in visual models. These experiments are conducted on the Ocean Water dataset which had 50 doubly-annotated visual models and an additional 94 single-annotated models. We evaluate our approaches first on the 50 models, then on the combined 144 models.

The same metrics are used for evaluation of both datasets. The "Dataset Accuracy" is calculated as the total number of true positive predictions divided by the total number of shapes in all images in the dataset. The "Average Accuracy per Image" is calculated by first dividing the number of true positives by the total number of shapes per image. Then, the resulting accuracies for all images are averaged. The motivation for having two accuracy measures is that the Dataset Accuracy is less sensitive to outliers in individual images. Since the goal of this project focuses on micro-object shape classification, the category of "Other" shapes will be excluded from all accuracy measures. Furthermore, Precision, Recall, and F1 Scores are calculated for each micro-object type in all visual models and are shown for the best two prediction methods, namely, the Rules-based approach and the Voting system.

The results in Table 3 show that the Rules-based Classification method achieves the best results for both datasets. The Decision Tree classifier is a close second on the 50-image subset with notably higher Average Accuracy per Image. Due to constraints with the shapes location annotations, machine learning was not applied to the 144-image dataset. The lower performance of the voting system implies that the Area Ratios and Hu Moments do not help improve the Rules-based approach in increasing the overall accuracy for this particular scenario.

To gain a clearer idea of why the Average Accuracy per Image is drastically lower than the Dataset Accuracy, each individual image's accuracy was analyzed for the Rules-based Classification. It was discovered that images with objects in a non-white background were achieving lower accuracy figures. This occurs due to our color segmentation process which merges ranges of colors for efficiency. However, the majority of the images were not influenced by this issue, which can be addressed in future work.

Table 3: Accuracy of the 50 and 144 Image Ocean Water Datasets.

	50 Image Ocean Water		144 Image Ocean Water	
Methods	Dataset Acc. (%)	Avg. Acc. Per Image	Dataset Acc. (%)	Avg. Acc. Per Image
Voting System	94.3	80.1	81.5	74.8
Rules-based	95.3	80.8	82.3	75.4
Hu Moment	90.2	77.9	74.7	70.0
Area Ratio	93.3	79.5	77.3	72.5
Decision Tree (Supervised)	95.0	94.0	-	-

Table 4: Voting System (5 Votes) on 50-image Ocean Water dataset

Voting System (5 votes), 50-image Ocean Water					
	Arrow	Circle	Square	Triangle	Diamond
P	95.7	93.5	78.6	96.0	100.0
R	90.7	94.7	94.3	88.9	100.0
F1	93.1	94.1	85.7	92.3	100.0
Rules-based Classification, 50-image Ocean Water					
	Arrow	Circle	Square	Triangle	Diamond
P	86.4	93.5	78.6	96.0	100.0
R	97.9	94.7	94.3	88.9	100.0
F1	91.8	94.1	85.7	92.3	100.0
Voting System (5 votes), 144-image Ocean Water					
	Arrow	Circle	Square	Triangle	Diamond
P	90.0	97.4	98.1	84.5	100.0
R	78.4	87.2	49.4	92.2	100.0
F1	83.8	92.0	65.7	88.2	100.0
Rules-based Classification, 144-image Ocean Water					
	Arrow	Circle	Square	Triangle	Diamond
P	90.6	97.6	95.7	94.7	100.0
R	84.2	87.4	49.7	92.2	100.0
F1	87.3	92.2	65.4	93.4	100.0

The resulting Precision, Recall, and F1 Scores for the Voting System with 5 votes and the Rules-based Classification are listed below in Table 4. The experiments are conducted on both the 50-image and the 144-image Ocean Water datasets. For the 50-image dataset, the Voting System had a higher F1 Score and Precision than the Rules-based Classification for Arrows, indicating that fewer False Positives were predicted. Since Arrow detection is important for modeling the Behavior sub-progression in any given LP, the improved scores for the Voting System validate its use. On the other hand, the Rules-based Classification achieves a higher Recall, indicating that fewer False Negatives are predicted. The rest of the shapes have identical scores, indicating the important role played by the Rules-based system in the Voting System.

A larger discrepancy emerges on the 144-image dataset. The Rules-based Classification has a higher F1 Score for Arrows, Circles,

Table 5: Quadratic Weighted Kappa for Datasets

	144-Image Ocean Water		174-Image Ocean Water		Two Can		50-Image Double-Annotated		
Learners System	**Voting-based**	**Rules-System**	**Voting-based**	**Rules-System**	**Voting-based**	**Rules-based**	**Voting-based**	**Rules-based**	**ML-based**
Regressors									
Linear	0.46	0.48	0.69	0.69	0.48	0.48	0.61	0.58	0.42
Decision Tree	0.52	0.50	0.65	0.64	0.22	0.22	0.57	0.57	0.42
Logistic	0.53	0.50	0.73	0.73	0.41	0.41	0.25	0.25	0.54
Random Forest	0.56	0.55	0.75	0.77	0.47	0.47	0.56	0.56	0.45
Support Vector	0.53	0.55	0.72	0.76	0.30	0.30	0.47	0.47	0.39
Classifiers									
Decision Tree	0.51	0.40	0.72	0.67	0.44	0.43	0.66	0.66	0.60
Random Forest	0.55	0.51	0.74	0.75	0.52	0.52	0.33	0.33	0.57
Support Vector	0.50	0.51	0.69	0.72	0.33	0.34	0.51	0.51	0.51

and Triangles indicating fewer false predictions, while the Voting System achieves higher precision for Squares. Notably, the F1 Scores for Squares are around 20% lower in the 144-image dataset for both methods. Further analysis showed that, in some cases, the background of the images were divided into parts that were incorrectly annotated by the human annotators as squares. Our approaches, however, were able to exclude them as they did not represent valid shapes. The high Precision, however, of 98.1% for the Voting System and 95.7% for the Rules-based system, indicates the reliable and accurate performance of our system.

6.2 Learning Progression Modeling

The ultimate goal of this project is to automatically assign visual models with a Learning Progression (LP) score between 1-4 using combinations of 10 aggregated features that were determined using the shapes features and classifications provided by our Voting and Rules-based systems. To predict the LP scores, the sets of aggregated features and their associated ground-truth LP scores are used in a 10-fold cross validation setting, i.e., 90% of the data is trained to generate predictions on the remaining 10% during each fold, and this iteration continues until predictions are generated for 100% of the data. Supervised regressors and classifiers are trained using the human annotated ground-truth scores to generate predictions by tuning on Quadratic Weighted Kappa (QWK), which measures the similarity between the predicted and human annotated scores on a scale of 0 to 1, where 0 represents no match and a score of 1 is an identical match. Additionally, Learning Curves of the Linear Regressions are displayed for each dataset to show the behavior of the error gap between training and validation sets as the number of training examples increases.

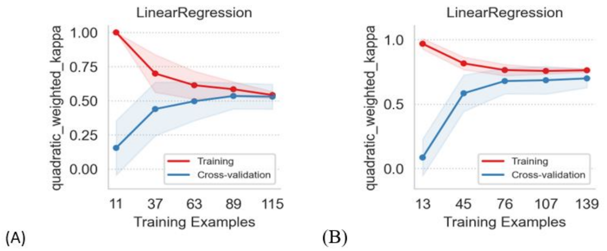


Figure 8: 144-image Ocean Water learning curves for (A) voting system and (B) Rules-based methods

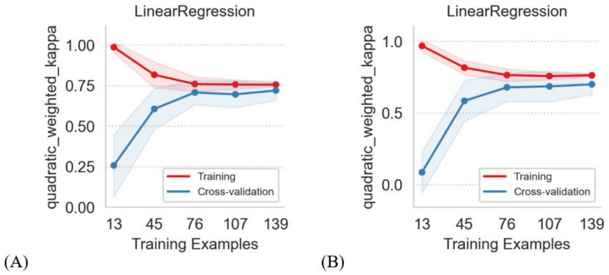
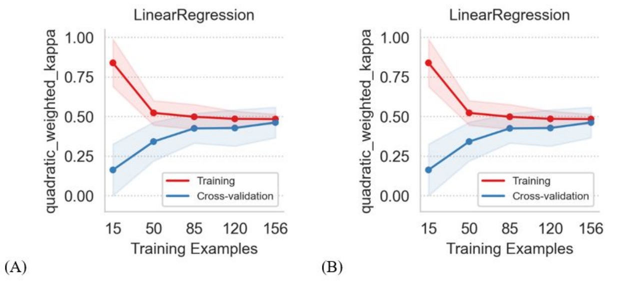


Figure 9: 174-image Ocean Water learning curves for (A) voting system and (B) Rules-based methods

The first dataset to be analyzed is the 144-image Ocean Water dataset. To compare the Rules-based Classification and the Voting System, both methods are independently used to extract all the necessary features. Then, the 10-fold cross validation is performed. The resulting average QWK scores for each type of learner are displayed in Table 5 and the corresponding Learning Curves are shown in Figure 8. The same analysis is applied to the 174-image Ocean Water dataset which includes the previous 144 models, along with 30 Synthesized Ocean Water models. Note again that these 30 models are synthesized due to a lack of such LP-4 models in the dataset collected from the students. The QWK scores for the larger dataset are displayed in the same Table 5 and the respective Learning Curves are shown in Figure 9.

The results for the 144-image Ocean Water dataset show a peak QWK score of 0.56 for the Voting System and 0.55 for the Rules-based system, both of which are achieved with the Random Forest Regressor, as seen in Table 5. For the 174-image Ocean Water dataset, an average QWK score of 0.716 with a standard deviation of 0.046 is achieved with the Rules-based approach and an average QWK score of 0.711 with a standard deviation of 0.033 is achieved with the Voting System. The peak score for both methods is achieved



(A) (B)

Figure 10: Two Can learning curves for voting system (A) and Rules-based (B)

using the Random Forest Regressor, achieving scores of 0.75 for the Voting System and 0.77 for the Rules-based Classification. These results indicate that it is possible to predict the LP level of an Ocean Water visual model instance using automatically extracted features, and such predictions agree well with human annotations.

The narrow error gap in both learning curves in Figure 9 also suggests a high bias situation, where adding more data for training probably does not contribute much to performance enhancement. Rather, efforts should be directed at conceiving and engineering more features to improve performance. Still, both approaches (Voting system and Rules-based) stagnated at around 0.76 for both training and cross-validation QWK suggesting that these methods are already minimally effective.

The second dataset to be analyzed is the 195-image Two Can dataset. Note that this dataset was not used to evaluate our shape detection systems, as it does not include human annotations of shapes. As earlier, the 10 aggregated features are extracted using the shape classification techniques provided by the Rules-based and Voting systems, followed by a 10-fold cross validation experimentation. The average QWK Score of the 10 folds is displayed in Table 5.

The resulting QWK Scores are very similar for both the Voting System and Rules-based approaches. This further shows that both of them generalize to novel datasets in a similar manner. For both methods, the average QWK Score for this dataset is 0.4 with a standard deviation of 0.1, while the peak score is achieved using the Random Forest Classifier with a QWK score of 0.52. These results are relatively lower than those of the Ocean Water dataset, a finding that can be best explained by the nature of the data in the Two Can dataset which contains hand-drawn objects and labels.

Further tests are conducted on the 50-image Ocean Water dataset to compare LP predictability using machine learning (supervised Decision Tree for shape classification in the first stage) for object classification and feature extraction with the Rules-based approach and the Voting System. The 50-image dataset is used as the ground-truth were doubly annotated for this subset, providing appropriate validation for supervised learning which requires a large number of valid labels. The results indicate that the Voting System and the Rules-based approaches are slightly better than the machine learning approach, as they achieve a QWK score of 0.66 compared to 0.6. Although the machine learning method achieves a lower QWK score compared to the other approaches, a score of 0.6 is promising and warrants future exploration of ML-based approaches for shape classification as additional data is acquired.

6.3 Cross-Domain Analysis

After testing the individual datasets, a cross-domain analysis is performed using the Ocean Water and Two Can datasets. Table 6 illustrates the results of using the most effective learners from Table 5, where the Ocean Water dataset is used for training and the Two Can dataset is used for testing, instead of the 10-fold cross validation used earlier with the individual datasets. The results show a peak score of 0.37 using Support Vector Regression. The average QWK score among the eight learners is 0.28 with a standard deviation of 0.065. These results are consistent with the theoretical expectations that cross-domain scores will be lower compared to *in-domain*, i.e., testing on the same dataset domain from which training instances are obtained.

Table 6: Cross-Domain Quadratic Weighted Kappa

	Ocean Water (train) **Two Can (test)**	**Two Can (train)** **Ocean Water (test)**
Learners	**QWK**	
Regressors		
Linear	0.21	0.47
Decision Tree	0.34	0.27
Logistic	0.25	0.49
Random Forest	0.34	0.23
Support Vector	0.37	0.21
Classifiers		
Decision Tree	0.26	0.12
Random Forest	0.28	0.16
Support Vector	0.19	0.21

The second cross-domain analysis is the reverse of the first. In this case, the Two Can dataset is used for training and the Ocean Water dataset is used for testing. Notably, there are 32 instances in the Ocean Water dataset that have an LP score of 4, whereas there are no instances of an LP score of 4 in the Two Can dataset. Hence, this LP score cannot be predicted as it is missing in the training data. For this reason, the 32 instances are removed, leaving a 144-image Ocean Water test set. In Table 6, the results show a peak QWK score of 0.49 for Logistic Regression, an average score of 0.27 for the eight learners, and a standard deviation of 0.137. The larger standard deviation in this case can be attributed to the presence of manually drawn shapes in the Two Can dataset, which is used for training, that are harder to detect correctly.

7 CONCLUSION AND FUTURE WORK

To automate scoring of scientific visual models drawn by students in the process of scientific knowledge acquisition, we proposed a multitude of image processing techniques, a novel cascaded system, an unsupervised Rules-based system, and a novel arrows orientation detector to classify different shapes and extract construct-relevant features. These classifications and features were aggregated into a set of 10 higher-level features used for effective scoring students' learning progression in a way that also addresses the scalability concerns.

The accuracy for shapes detection and classification achieved a maximum of 95.3% on the 50-image dataset that has doubly annotated labels with our Rules-based system. Our approaches, along with the novel method to determine Arrows orientation, were validated by the high QWK scores, averaging above 0.7 for the 174-image Ocean Water dataset and peaking at 0.56 for the 144-image dataset. This indicates the feasibility and potential of employing a fully automated pipeline to effectively score visual models.

A key aspect that can be explored for future improvement without additional resources is color segmentation. Upon further analysis, it was found that few images had objects in a non-white background. Our belief is that during the binarization of the image, the objects were mixed in with the background, thus making them undetectable. Future work can explore methods to remedy this.

The results of automated scoring using Rules-based approach and the Voting System indicated the feasibility of applying our novel approaches for automatic scoring to solve the scaling issue and show promise for this new line of research, making scientific visual modeling a viable tool for widespread use.

ACKNOWLEDGMENTS

This material is based in part upon work supported by the Educational Testing Service (ETS). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ETS.

REFERENCES

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. 2009. From contours to regions: an empirical evaluation. *Proceedings of International Conference on Computing Vision and Pattern Recognition (CVPR)* (2009), 2294–2301.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. 2011. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal Mach. Intell.* 33, 5 (2011), 898–916.

[3] M. Basu. 2002. Gaussian-based edge-detection methods—a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 32, 3 (2002), 252–260.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. 2008. Speeded-up robust features (SURF). *Computer vision and image understanding* 110, 3 (2008), 346–359.

[5] D. Bremner, E. Demaine, J. Erickson, J. Iacono, S. Langerman, P. Morin, and G. Toussaint. 2005. Output sensitive algorithms for computing nearest neighbor decision boundaries. *Discrete and Computational Geometry* (2005), 593–604.

[6] C. A Burgers. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2, 2 (1998), 121–167.

[7] J. A Canny. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, 6 (1986), 679–698.

[8] T. Corcoran, F. Mosher, and A. Rogat. 2009. *Learning progressions in science:An evidence-based approach to reform.*

[9] National Research Council et al. 2013. *Next generation science standards For states, by states.*

[10] National Research Council et al. 2014. *Developing assessments for the next generation science standards.* National Academies Press.

[11] T. Cover and P. Hart. 1967. Nearest-neighbor pattern classification. *Information Theory IEEE Transactions* (1967), 21–27.

[12] Q. Deng and Y. Luo. 2011. Edge-based method for detecting salient objects. *Optical Engineering* 50, 5 (2011), 301–301.

[13] D. Douglas and T. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a line or its character. *The American Cartographer* 10, 42 (1973), 112–123.

[14] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G Aref. 2001. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Trans. Image Process* 10 (2001), 1454–1466.

[15] J. Flusser and T. Suk. 1993. Pattern recognition by affine moment invariants. *Pattern Recognition* (1993), 167–174.

[16] J. Flusser, B. Zitova, and T. Suk. 2009. *Moments and Moment Invariants in Pattern Recognition.* Wiley Publishing.

[17] K. Forbus, J. Usher, A. Lovett, K. Lockwood, and J. CogSketch Wetzel. 2011. CogSketch:Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science* 3, 4 (2011), 648–666.

[18] M. K. Hu. 1962. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions* 8 (1962), 179–187.

[19] E. Karami, S. Prasad, and M. Shehata. 2015. Image Matching Using SIFT, SURF, BRIEF, and ORB: Performance Comparison for Distorted Images. *Proceedings of the 2015 Newfoundland Electrical and Computer Engineering Conference St. John's, Canada* (November 2015).

[20] J. Kim, B. S. Kim, and S. Savarese. 2012. Comparing image classification methods: K-nearest-neighbor and support-vector-machines. *Proc. of 6th WSEAS Int. Conf. on Computer Engineering and Applications, and Proc. of the 2012 American Conf. on Applied Mathematics* 6 (2012), 133–138.

[21] C. W. Leong, L. Liu, L. Chen, and R. Ubale. 2018. Toward Large-Scale Automated Scoring of Scientific Visual Models. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale.*

[22] Y. W. Lim and S. U. Lee. 1990. On the color image segmentation algorithm based on the thresholding and the fuzzy C-means technique. *Pattern Anal. Mach. Intell.* 23, 9 (1990), 935–952.

[23] T. Lindeberg. 1998. Edge Detection andRidge Detection with Automatic Scale Selection. *Int'l J. Computer Vision,* 30 (1998), 117–156.

[24] L. Liu, R. Rogat, and M. Bertling. 2013. *A CBAL Science Model of Cognition: Developing a Competency Model and Learning Progressions to Support Assessment Development.* ETS R-R Report.

[25] D. G. Lowe. 1999. Object recognition from local scale-invariant features. *IEEE Int Conf. Computer Vision* 2 (1999), 1150–1157.

[26] N. Madnani and A. Loukina. 2016. RSMTool collection of tools building and evaluating automated scoring models. *Journal of Open Source Software* 1, 3 (2016), 33.

[27] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. 2008. Using contours to detect and localize junctions in natural images. In *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR).* 1–8.

[28] D. Martin, C. Fowlkes, and J. Malik. 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 5 (2004), 530–549.

[29] J. D. Merritt. 2010. *Tracking students' understanding of the particle nature of matter.* Ph.D. Dissertation. University of Michigan.

[30] M. A. Oskoei and H. A Hu. 2010. *Survey on Edge Detection Methods.* Technical Report. University of Essex ,Technical Report: CES-506.

[31] N. Pal and S. Pal. 1993. A review on image segmentation techniques. *Pattern Recognit.* 26 (1993), 1277–1294.

[32] G. Papari and N. Petkov. 2011. *Edge and line oriented contour detection: state of the art.* Image and Vision Computing.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas. 2011. Scikit-learn Machine learning in Python. *Journal of machine learning research* (2011), 2825–2830.

[34] J. M. S. Prewitt. 1970. Object enhancement and extraction Picture Processing and Psychopictorics. B. Lipkin and A. Rosenfeld A. Rosenfeld Academic Press, New York.

[35] L. G. Roberts. 1965. Machine perception of three-dimensional solids. In *Optical and Electro-optical Information Processing*, J. T. Tippet (Ed.). MIT Press, 159–197.

[36] E. Rublee, V. Rabaud, K. Konolige, and G. Orb Bradski. 2011. An efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision.*

[37] Stephanie AC Ryan and Mike Stieff. 2013. Using multiple modalities simultaneously as an assessment tool for learning from visualizations. In *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY*, Vol. 245. AMER CHEMICAL SOC 1155 16TH ST, NW, WASHINGTON, DC 20036 USA.

[38] J. Seo, S. Chae, J. Shim, D. Kim, C. Cheong, and T. D. Han. 2016. Fast contour-tracing algorithm based on a pixel-following method for image sensors. *Sensors* 16, 3 (2016), 353.

[39] M. E. Sobel. 1982. Asymptotic confidence intervals for indirect effects in structural equations models. In *S.Leinhart Sociological methodology.* 290–312.

[40] S. Suzuki and K. Abe. 1985. Topological Structural Analysis of Digitized Binary Images by Border Following. *CVGIP* 30, 1 (1985), 32–46.

[41] C. H. Teh and R. T. Chin. 1988. On image analysis by the methods of moments. *IEEE Trans. Pattern Anal Machine Intell.* 10 (1988), 496–513.

[42] L. A. Torres-Mendez, J. C. Ruiz-Suarez, L. E. Sucar, and G. Gomez. 2000. Translation Rotation and Scale-Invariant Object Recognition. *IEEE trans.on systems, man and Cybernetics* 30, 1 (2000).