The paper "Extracting Features for Computational Thinking from Block-Based Code" by Param Damle, Jo Watts, Glen Bull, and N. Rich Nguyen addresses the challenge of supporting undertrained instructors in introductory computer science classes by developing an automated evaluator (autograder) for Snap!, a block-based programming language. The autograder aims to assess computational thinking (CT) by evaluating the structure of student submissions, focusing on concepts like abstraction and iteration rather than traditional metrics like accuracy or run-time performance. The authors explore feature extraction tailored to capturing CT elements, such as repetition and encapsulation, from an XML tree representation of Snap! programs. Their approach integrates the academic community into the research and development of the autograding model to enhance its effectiveness and scalability for diverse classrooms.

The study involves creating a rubric to evaluate Snap! programs, where students' coding assignments are transformed into numerical scores reflecting various CT skills. The researchers design features to measure structural aspects of the code, including the average number of children per parent node and the depth of nodes. They also develop a 1-dimensional convolution method to assess code repetition, which helps identify redundant sections that could benefit from abstraction. Various clustering algorithms, such as K-means, DBSCAN, Gaussian Mixture Models, and Hierarchical Clustering, are tested to group similar submissions based on these features. The performance of these models is evaluated using relative inertia, a metric that compares clustering error to human-assigned scores.

The results indicate that Gaussian Mixture Models and Hierarchical Clustering show promise in clustering submissions according to the manual rubric. However, the study highlights the limitations of current autograding models, such as the inability to provide specific feedback on areas of improvement or appreciate the artistic aspects of multimedia programs. The authors suggest future work to improve feature embedding and potentially incorporate synthesized text tools like ChatGPT for personalized feedback. The study underscores the need for an academia-facing autograder for block-based languages that balances scalability and educational effectiveness.