Evaluating clustering methods underpinning content generation in games using GANs

Gabriel Lacey
Ruth E. Falconer

This is the Author Accepted Manuscript of a conference paper published in GAME-ON'2020: proceedings of the 21st International conference on Intelligent Games and Simulation.

Lacey, G. & Falconer, R.E. (2020) 'Evaluating clustering methods underpinning content generation in games using GANs'. In: A. Veloso, O. Mealha & L. Costa (eds.) *GAME-ON'2020: proceedings of the 21st International conference on Intelligent Games and Simulation*. EUROSIS, 21st annual European GAMEON® Conference, Aveiro, Portugal, 24-25 September 2020

EVALUATING CLUSTERING METHODS UNDERPINNING CONTENT GENERATION IN GAMES USING GANs

Gabriel Lacey
Ruth E Falconer
School of Design & Informatics
Abertay University
DD1 1HG, Dundee,
Scotland
E-mail: glaceydev@gmail.com

KEYWORDS
Artificial Intelligence, Clustering, Computational creativity, Generative Adversarial Networks, Image processing

ABSTRACT

In recent years there has been a push for more customisation options in games, along with a desire for greater realism. While graphics have been steadily improving year over year, the current customisation options remain limited, however thanks to developments in research surrounding generative artificial intelligence the combination of both of these desires may be made possible through the use of the latest Generative Adversarial Networks. The aim of this project is to implement and compare four different clustering methods. These methods will be used to generate classification labels from gameplay images which will then be given as input to a generative network to create photorealistic equivalents. It will then be determined which method is most suitable for this task by comparing their initial classification performance and the results from the photorealistic images they are used to generate. In order to compare classification performance, the Dice coefficient was calculated for each classification image generated, using a ground truth image to represent perfect segmentation. It was found that good classification performance does not necessarily lead to superior GauGAN output images, and overall the best performing method for this task was Region-Growing due to the spatial consideration in its approach.

INTRODUCTION

Customizing and sharing experiences is a fundamental part of how we enjoy games and interact with each other – even before video games were a mainstream form of entertainment, tabletop games such as Dungeons and Dragons thrived by allowing players to make the game their own with custom campaigns, characters, and enemies to be shared among a group of friends. Even now these aspects retain the popularity of more traditional games, while also making their way into modern video games in the form of fan-made modifications (mods).

At the same time as video games have evolved to allow for aspects such as these, there has also been a push for higher realism in every aspect of gameplay – from mechanics to sound design, this theme has been persistent as a sign that increased realism makes games more immersive and enjoyable for certain player demographics. Part of this trend has seen the graphics of games gradually tend towards photorealism within some genres, as people want to see as close to a real life equivalent of what they are experiencing in game as possible.

These ideas of customization and an increased desire for realism forms the basis of the entire project which attempts to convert stills of games using different unsupervised learning clustering methods into segmentation maps describing the class of every object present on a per-pixel basis, a level of detail is required for use in the next stage as input for a Generative Adversarial Network (GAN) to produce a photorealistic image with the same objects and context provided by the game. These clustering methods will then be compared against each other in terms of pure clustering performance along with their applicability to this specific context, with a final recommendation on which method is likely to give the best results based on results gathered.

METHOD

The project itself can be split into two main stages – image processing and GauGAN translation. The vast majority of the implementation work falls under the former, which itself can be split into three sub-stages. Data loading and preparation is the simplest of these, in which the images to be used throughout the rest of the program are loaded in and have any pre-processing operations applied to them, such as image resizing or smoothing. While it does not involve much work, this step has a knock-on effect to the rest of the application – the main consideration in this area is any downsizing applied to the source image as this can benefit the performance of the application, typically at the cost of the accuracy of results due to lower resolution segmentation map outputs. The latter aspect of GauGAN translation is the least involved practical part of the project and simply involves uploading the resultant segmentation maps to the nVidia GauGAN tool online. These maps are then processed by the tool into a photorealistic output with a number of styles to choose from, selected based on which gives the most accurate output at the time.

Colour Difference

Colour comparison is a tricky area of computer vision in that the usual way numbers are compared for similarity do not match up with a human's perception of colour. For example, an intuitive way to compare two points in a three-dimensional system, such as RGB 'coordinates', would be to find the Euclidean distance between the two points, however this approach can quickly encounter problems due to the

failing of this method to take different shades of colours into account when comparing them and treating intensity as an equal part of the calculation. As a solution to this, the CieLab (LAB) colour space (Luo, 2015) was developed as a different way to represent colours – instead of the typical red, green, and blue channels colours are represented by their lightness (L), the colour on a scale from green to red (A), and the colour on a scale from blue to yellow (B). This allows comparison calculations to put emphasis on different parts of a colour's properties, in this case putting less emphasis on a colour's intensity. Alongside the LAB colour space, a series of 'Delta E' formulae were developed to utilize this different representation of colour, gradually getting more sophisticated with each iteration. In this application these two Delta E formulae will be used wherever a substantial advantage is given by either of the two, and where the difference between both is negligible Delta E 76 will be used due to its better processing time. The choice made for each clustering method is detailed in the following section.

Clustering Methods

The implementation of the clustering methods themselves forms the most crucial part of the application as it is here that the accuracy of the data that the classifier must work with is determined, and therefore the accuracy of the application results overall. Each of the four methods implemented were selected for their different core approaches to segmentation in a hope to draw out different strengths and weaknesses potentially making them more applicable for different scenarios.

The first of these four is the K-Means algorithm, selected due to its relative simplicity making it a good basis for other algorithms to be compared to. It functions by taking a set number of target clusters as input and gradually determining the location of these clusters within the colour space by iteratively updating each pixel in the image to the closest cluster available. These clusters then have their centres calculated as the average value of all member pixel's colours, and the process is repeated until convergence is reached.

The next method, Mean-Shift, takes a different approach in developing its cluster centres by employing a density-based method to find the local maxima within the dataset. It does this by initially setting each pixel as a cluster centre and then gradually moving them towards dense areas within a local window until they are either combined with another cluster or converge on the most dense area within their local window. This provides the advantage over mean shift of not needing to define the number of clusters initially, however the radius used for the local window around cluster centres must still be defined.

Region-Growing takes a completely different approach in comparison, considering each pixel individually and adding them permanently to their final clusters. It does this by assigning a number of seed pixels as the initial regions distributed evenly across the image. Each of these clusters is then considered in turn comparing each member pixel to its neighbouring pixels and adding them to the cluster if the new pixel is similar to the initial seed pixel of the group using a predefined similarity factor. This process continues until either all pixels belong to a region or no more pixels can be added to regions, at which point new seed pixels are assigned and the process starts again.

Simple Linear Iterative Clustering (SLIC) was developed in 2010 (Achanta, R. et al., 2010), specifically for the task of image processing and starts similarly to Region-Growing by initializing seed pixels as clusters across the image. The primary difference is that a cluster only considers pixels within a certain radius around its position, making this method far more spatially considerate than any others tested. To calculate the membership of a pixel between certain clusters, the similarity in colour and the physical distance between the pixel and cluster centre are combined into a five-dimensional similarity formula. Over the process of convergence, neighbouring clusters swap pixels back and forth while their centre positions are updated until no more swaps are made.

Classification and GauGAN

Classification forms the final stage of this application where an attempt is made to determine the objects and features present in a segmentation map on a pixel-by-pixel basis. The first step in this process is loading in the data describing the target classes and their corresponding features – in this case class information has three components loaded: the average colour of every texture in game, the colour of the related classes in the GauGAN application, and the name of these classes from a text file. The texture and label information are stored as two separate image files, with the relation between each being defined as their position in the images themselves (figure 1).

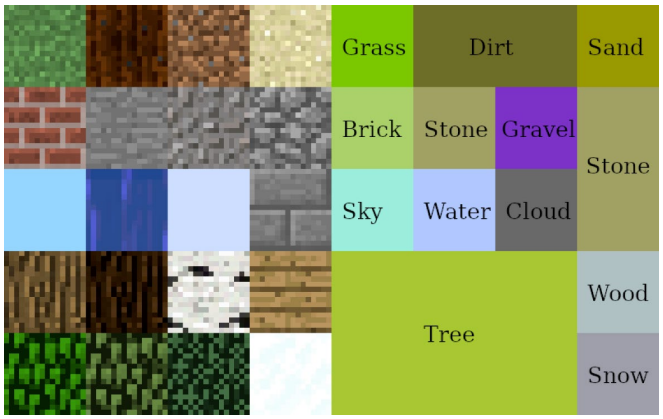


Figure 1 - Classification labelling data

The method then goes through every pixel in the segmented image, comparing the colour of each to the texture colour of every loaded class after which the corresponding pixel in the output image has its colour set to the paired label colour of the closest texture measured. The accuracy of this stage is heavily influenced by the method of colour comparison as mentioned previously, so to maximise performance different outputs are generated each using a different comparison method. These are then evaluated separately with the highest scoring method being retained. The images generated by the classification stage use the same colours to describe each label as those used in the GauGAN application, so no additional processing is needed, and the images can be directly uploaded to the online application. As part of this upload process images are automatically resized from the resolution of the screenshot taken in game to the much lower

resolution used for the model. Once uploaded an image can be converted into multiple different photorealistic outputs using style filters available – the style used will vary between scenarios depending on which produces the best results with the ground truth image. Once processing is complete, the photorealistic output can then be downloaded for comparison alongside the other generated images.

Testing and Evaluation

The four clustering methods implemented will be tested and evaluated . The first and most important area of comparison is how well these methods classify a given image, using the Sørensen–Dice coefficient (Sørensen–Dice coefficient, 2020)–It calculates a region's score by comparing the area predicted by the clustering attempt (Ap), the area of the same class in the Ground-Truth image (Al), and the area of overlap between these two (Apl) as seen in equation 1. Consideration must also be given to the testing scenario and seven scenes were selected based on the scene complexity and lighting effects. Seven scanarios with very different characteristics were identified for the evaluation e.g Forest, Snow, Day, Canyon, Underground, Beach and Night scene.

$$DSC = \frac{2Apl}{Ap + Al}$$

Equation 1 - Sørensen–Dice coefficient

RESULTS

Each clustering method has at least one user defined parameter which influences the accuracy. Instead of using one fixed value for all testing scenarios, which would favour some scenarios over others, each method was tested for each scenario extensively with a wide range of parameters to find the best-case performance for each scenario and method combination. These are presented in Figure 2.

Figure 2 Variance in 'best' parameters for each clustering methods across scenarios

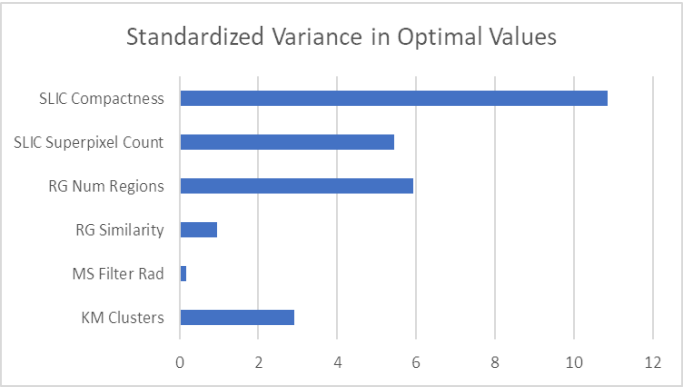
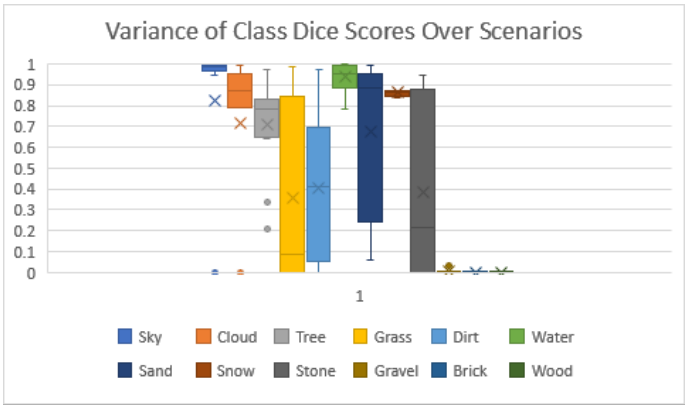


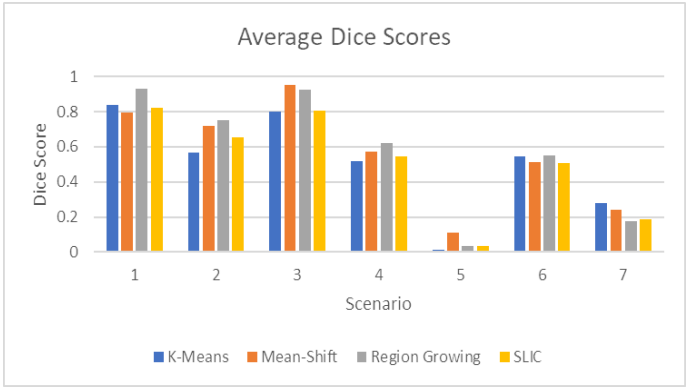
Figure 3 shows the diustribution of Dice scores across scenarios and classes (sky, sand, stone, water, wood). A Dice score is calculated for each region present in the Ground-Truth image across all seven scenarios. The sky and water classes are more accurately classified. The most variance appears in the stone, grass and brick classes.

Figure 3 Performance of classification scores across classes and scenarios.



An average score will also be taken from the average of all region scores within each image and to give an indication to the overall performance of methods in each scenario, with each individual region having equal weighting towards this final score (Figure 4).

Figure 4 Performance of classification scores across scenarios and clustering methods



The visual result for scenario 1 Forest is presented in Figures 5 - 9 showing the game source image, clustering and labelling result based on KMeans, Mean-shift, Region-growing and SLIC, and GauGAN output of that clustering and classification.

Figure 4 Scenario 1 – K-Means

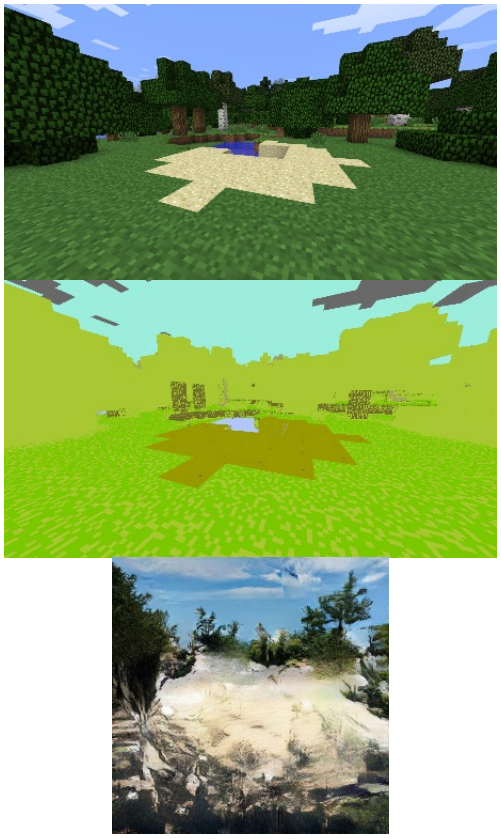


Figure 5 Scenario 1 – Mean-Shift

Figure 6 Scenario 1 – Region-Growing

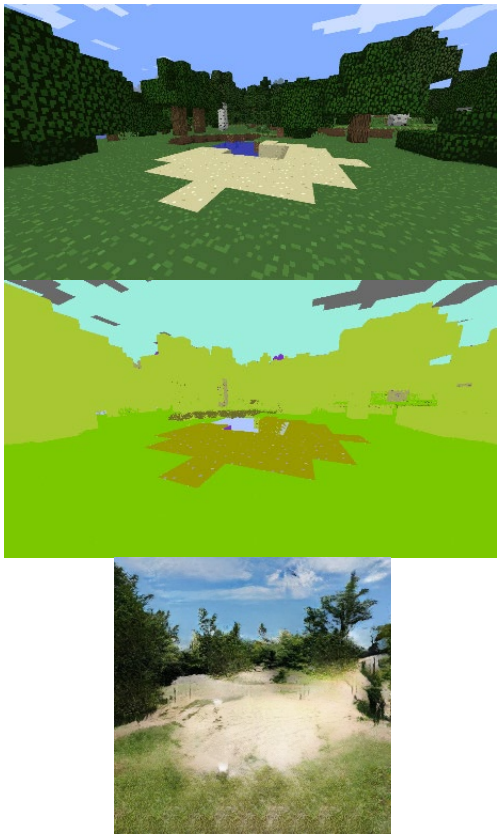


Figure 7 Scenario 1 – SLIC

CONCLUSION

The primary goal of combining unsupervised learning and Generative Adversarial Networks for the purpose of computational creativity in games has been demonstrated. The clustering methods display varying results across different criteria that make the methods applicable to different use cases, and as it stands Mean-Shift appears to be the most reliable method in terms of pure clustering performance. As demonstrated however this does not necessarily carry over to desirable GauGAN results due to issues of noise, and as such the local, spatially based methods of Region-Growing and SLIC are more suitable for this specific task. Of these two methods, Region-Growing is recommended as the overall best solution for clustering due to its high classification performance when compared with all other methods. With the further work suggested however this recommendation may be subject to change as there are options to resolve each method's shortcomings – in this regard any method may become performant enough to justify its use.

The other factor for consideration is the appropriateness of GauGAN for the task of generating images. From the beginning of the project building a custom network was out of scope, and as such GauGAN became the default option due to its ease of access and, most importantly, its superior performance compared to the previous networks it was based off. Despite this limitation GauGAN performed well and served as a good base to test the hypothesis without much obstruction. It is clear however from the results that many improvements could be made to better fit the network by fine-tuning parameters. For example retraining the SPADE network for every novel setting/scenario would lead to improved results. In addition, out with the core focus of this project, this method may also have potential not just for content creation on finished games but also as a tool for artists and designers to use during the production of the game itself. To accomplish this a model could be trained on images from the game to generate new scenarios from its existing style and could be used to allow creators to mock-up new level designs and setting concepts.

This study also demonstrates the potential for the existing creative effort put into developing a game to be utilized for Computational Creativity. The visual methods discussed in this paper only highlight one aspect of this potential as well, with areas such as sound and level design possibly opening entirely new routes of investigation. On the other hand, the results obtained also demonstrate the diversification Computational Creativity can offer to content customisation in games as the photorealistic images generated are unlike anything available in current games. All of this serves only as a starting point of interest, with improvement required in almost every area of the process before such methods become commercially viable and ready for mass inclusion in games, however with further integration to a game's content pipeline this improvement should be attainable without much more work required.

Overall this project may be considered a success in its initial goal of combining unsupervised learning and generative networks to create entirely a new way to visualise existing game worlds and presents several future lines of investigation.

FUTURE RESEARCH

While the current implementation of this project serves well as a proof of concept, further work would be required to realise it as a commercially viable application. Part of the limitation comes down to the unreliable classification performance which would need to be addressed by the improvements discussed above, or to implement a different method of segmentation which overcomes the shortcomings of unsupervised learning. Fortunately, this task may be trivial in a games application when developing the game itself since perfect segmentation performance could be achieved by building the functionality directly into the game's graphics pipeline, which would work by simply changing the material of every object to a solid colour of its intended label and disabling any additional lighting and effects. Without this type of access to a game's source code, machine learning methods, such as those discussed in this paper, can serve as a good alternative for more limited application. The GauGAN model used for this project is also limited in that it can only generate images of landscapes from the datasets it has been trained on, as has been discussed from the results gathered. It would likely then be beneficial to train a bespoke model for games application, particularly if the intended use case is a fiction or fantasy setting as many of the labels that would be identified in these will not have real-life equivalents. To accomplish this, it would then be necessary to create a custom dataset with photorealistic fictional settings and their labelled equivalent, perhaps sourced through live action films and television series. This use case would be particularly applicable to games that have a direct relation to such media. For example, the game Star Wars Jedi: Fallen Order (EA, 2020) would have a vast amount of live action source material from the Star Wars franchise meaning that the majority of objects, characters, and settings in the game could have a 'real-life' equivalent. A further benefit in training a custom generator would be in the ability to adapt specifically how the model is trained based on the use case, making it possible to train a network with more "creative freedom" in how it deals with the input labels. For example, a network could be trained to accept input as a description of the rough location of a class within an image, rather than the pixel-wise implementation currently used. Additionally, the model could be adapted to accept more general labels to allow it more room to work with when generating the photorealistic equivalent of the region – for example trees and bushes could instead be described simply as "foliage". This would likely result in an output that would be less true to the source image, however it could appear more realistic overall due to more emphasis put on this factor.

REFERENCES

Achanta, R. et al. (2010) 'SLIC Superpixels', EPFL Technical Report 149300

Luo, M., 2015. CIELAB. Encyclopedia of Color Science and Technology, pp.1-7.

WEB REFERENCES

EA, 2020. Star Wars Jedi: Fallen Order™. [online] Electronic Arts. Available at: <https://www.ea.com/en-gb/games/starwars/jedi-fallen-order> [Accessed 22 April 2020].

En.wikipedia.org. 2020. Sørensen–Dice Coefficient. [online] Available at: <https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient> [Accessed 22 April 2020].

BIBLIOGRAPHY

Gómez de Silva Garza, A. (2019) 'An introduction to and comparison of computational creativity and design computing', *Artificial Intelligence Review*. Springer Netherlands, 51(1), pp. 61–76. doi: 10.1007/s10462-017-9557-3.