Detecting Gerrymandering with Deep Learning Models

Anonymous Author(s)
Submission Id: 519

ABSTRACT

Gerrymandering is an issue of paramount importance in several parts of the world, particularly in the US, and has been the subject not only of a significant amount of literature, but also of recent Supreme Court cases and official redistrict plans. It poses a great threat to democracy, and the need for the development of appropriate detecting and preventing measures is evident. In this paper, we employ machine learning as a predicting measure for gerrymandering. In particular, via combining appropriate image district data and their associated demographic data, we train deep learning models that are able to detect gerrymandered districts with rather high accuracy. Our results show for the first time the potential effectiveness of deep learning as a tool against gerrymandering, and opens up avenues for further investigations.

KEYWORDS

gerrymandering, elections, deep learning, image recognition

ACM Reference Format:
Anonymous Author(s). 2023. Detecting Gerrymandering with Deep Learning Models. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 7 pages.

1 INTRODUCTION

Gerrymandering [14], or the geographic manipulation of voters to sway election results, in modern democracies is a challenging issue to address and quantify. It occurs when policymakers draw voting districts, within a larger state or country, in such a way that a political party wins more proportional representation in the legislative body than their popular vote share indicates. For example, in a two-party system, a gerrymandered election would look similar to the results in Table 1. Party A received a minority of the popular vote, but still has majority control in the legislature. Party A achieves this gerrymandered result by a combination of "packing" party B's supporters into fewer districts and "cracking" concentrated areas of party B supporters across multiple districts, thus diluting their vote [17, 31]. The practice is illegal and/or mitigated in most democracies via non-partisan drawing commissions.

However, in the absence of appropriate measures to curb the effects of gerrymandering, the consequences might be quite severe. As Issacharoff [2002] argues, gerrymandering causes three central damages: harm to democracy through loss of democratic accountability, harm to the individual through weakening of an individual right (voting), and harm to specific minority groups via group-based discrimination. As citizens experience these adverse effects of gerrymandering, they begin to lose faith in their election systems. Consequently, if this cynicism and these infringements become severe enough, democracy itself begins to erode, leading to public unrest and at times even radical responses.

Table 1: An example of gerrymandered election results

	Party A	Party B
Popular Vote Share	49%	51%
Legislative Share	55%	45%

Although gerrymandering is technically illegal in most democracies, it still persists in several areas of the world, most notably in the United States of America. This is due to the fact that gerrymandering is often quite subtle, and there is a seeming lack of widely accepted systematic approaches for detecting it. The most common methods are statistical – developing metrics and indices to find outliers in a set of districting plans. Among those, most notable is the work of Bangia et al. [2017], which was used as a central source of evidence in the recent "Rucho vs Common Cause" US Supreme Court case [2], see also [30]. Ultimately however, the US Supreme Court ruled that the issue of partisan gerrymandering was not justiciable by the federal courts due to a lack of a "limited and precise standard for evaluating partisan gerrymandering" [2].

The origins of the term "gerrymandering" trace back to the early 1800s and the actions of Elbridge Gerry, the then Governor of Massachusetts, who signed a bill in which the shape of a partisan district in the area of Boston resembled that of a salamander [14]. Besides an interesting anecdote, the underlying principle of this fact is that Gerry's districting plan did not pass "the eye test". Of course nowadays gerrymandering is much more sophisticated, but so are the "eye tests" that the community has at its disposal. In particular, the emergence of deep learning [20] in recent years has revolutionized the area of image and pattern recognition (e.g., see [15, 33]), surpassing the capabilities of humans [23]. Given this tremendous success, it is natural to ask whether a similar approach could actually be used to detect gerrymandered states, by simply "looking at them," via the lens of meticulously developed and trained deep learning models, or at least via an appropriate combination of district images and some associated demographic data. This is the main focus of this work; to the best of our knowledge, we are the first to provide affirmative evidence in this direction.

1.1 Our Contributions

We train deep learning models to detect gerrymandering in districting plans. In particular, our models input image data (i.e., images of the districts) and their associated demographic data (e.g., the proportions of certain populations in the district) and classify the district as being gerrymandered or not. In more detail:

- We generate 27,900 images of realistic congressional districts based on actual real districting plans from the US Census Bureau [1], given a set of appropriately chosen constraints.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

We perform cleaning and preprocessing to the image data to make them amenable to training for our models.

- We build a deep learning model that combines a convolutional neural network (CNN) for the image data with a simpler network with fewer hidden layers for the demographic data, for both a binary classification task (i.e., detecting whether districts are gerrymandered or not) and a multiclass classification task (i.e., detecting the extent of gerrymandering as low, medium, or high).
- We show that our trained model reaches a high level of accuracy, i.e., as high as 86% for the binary task of detecting gerrymandered states.

To the best of our knowledge, this is the first time that deep reinforcement learning has been used for detecting and predicting gerrymandering. While our findings leave open some key avenues for future investigation, they clearly demonstrate the effectiveness and the potential of the technique for this important and challenging task.

1.2 Related Work

As we mentioned in the Introduction, the issue of gerrymandering was highlighted as early as the 19th century [14], and has assumed a prominent role in the political science literature, as well as social science, mathematics, economics and recently also computer science [6–8, 13, 18, 22, 25, 26, 28, 32].

As we explained earlier, Bangia et al. [2017] offered a statistical analysis with many visualizations as evidence of gerrymandering for the recent Supreme Court case of "Rucho vs Common Cause" [2]. To generate the data for this analysis, the authors used a Monte Carlo Markov Chain (MCMC) method to generate thousands of realistically drawn districts for North Carolina. We used the same MCMC generation approach in this work; we provide the details in Section 2.1. Following the generation of districting plans via the MCMC, Bangia et al. [5] implemented metrics and statistical tests to quantify the effect of gerrymandering across the plans, and compare them to the actual plans used in North Carolina's elections. They concluded that the actual plans were gerrymandered outliers according to the metrics they used and were skewed to favor the Republican party.

The work of Bangia et al. [5] along with the related work of Herschlag et al. [2017] offer an accurate summary of the current methods and sophistication of gerrymandering detection. These papers also highlight several metrics to quantify and measure gerrymandering which were of particular interest for our work. Perhaps the most prominent of them is the *efficiency gap score (EGS)* metric that we use as the metric for gerrymandering in our experiments, see Section 2.2 for the details. Other metrics proposed are the *Gerrymandering Index* and the *Representative Index* [5, 16]; the first is rather simple mathematically while the second, which is more conceptual, has been shown to correlate highly with the EGS [5]. For this reason, we adopted the EGS as the gerrymandering metric for our work.

A notable line of work in computer science and artificial intelligence has focused on how the parameters of a district or state (e.g., population density, district size, number of total districts) impact, nullify, and/or amplify gerrymandering. For example, Borodin et al. [2018], analyzed the effect of the urban vs. rural population dynamic in gerrymandering and asserted that the preferred party of the rural voters has more gerrymandering power. The authors, in fact, developed a Mixed Integer Linear Program (MILP) to generate gerrymandered districts. The algorithm tries to maximize a party's representation above their proportional share, which is the sole metric that they use to detect gerrymandering. Lewenberg et al. [2017] also use a generated grid-state, but contrary to [7], they analyze just how effective gerrymandering can be for parties and the parameters that impact it. They confirm why gerrymandering is such a concern for democracies by providing two real world examples (2015 Israeli and British elections) and illustrate how different district parameters and compositions can lead to any of the major parties involved winning pluralities. The use of the generic grid-states (as opposed to real US states) as the input districting data allowed the authors to make more generalized models and conclusions about gerrymandering. While generalized models are useful, our goal is to remain as grounded in reality as we could for this analysis. The assumptions made in those works often do not take important US federal laws on drawing districts into account (e.g., district populations must be within a certain percentage of each other), and, in turn, add a layer of distortion between the model and its applicability to US politics.

Finally, we remark that computer-aided solutions for detecting gerrymandering have been proposed in the past (e.g., [9, 10]) but primarily in the form of simulations, and certainly markedly different from the deep learning approach that we advocate in this work.

2 EXPERIMENTAL SETUP

2.1 Data Generation

For the data generation, ideally we would like to use real districting plans from elections in the US. However, given that redistricting typically happens only once in a significant number of years (typically one decade, see [21]), that would result in a rather restricted dataset that would be insufficient for training and learning.

Realistic Districts via GerryChain. The most reasonable alternative route is to generate realistic instances of districting plans for states, given an initial set of real-world districting plans. For this, we employed *GerryChain* [3], a Python library developed by the Metric Geometry and Gerrymandering Group[1], which enables the creation and tracking of realistic districs and accompanying demographic data. At a high level, the library employs the Monte Carlo Markov Chain (MCMC) to algorithmically flip groups of border voting districts to create new congressional districts. Given a starting map with congressional districts defined by smaller voting districts, the algorithm will generate new congressional districts subject to constraints provided by the user.

More technically, the MCMC uses the Metropolis-Hastings algorithm to "define a random walk" on the set of all possible redistrictings "which has [a probability distribution] as its unique, attracting stationary measure" [5]. Using this algorithm it selects a group of border voting districts within a congressional district

[1] https://mggg.org/

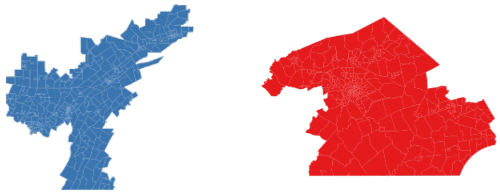

Figure 1: An example of generated congressional districts.

and flips them to the adjacent congressional district. It then verifies whether all of the newly generated congressional districts are within the provided constraints, and if so, it saves the plan and repeats the process with a new set of voting districts. If the plan falls outside of the constraints, the algorithm will undo the flip of voting districts and attempt to flip a different set of voting districts. The constraints that one can set include in the district generation are population deviations, non-contiguous congressional districts, and district compactness. More information about the engine of the chain, the Metropolis-Hastings algorithm, can be found in Robert [2016].

Real-world data. The real-world map and associated census data utilized for this analysis was originally collected from the US federal government, specifically the Census Bureau [1]. The map data is comprised of folders that contain several files with geospatial IDs and geodata that allow maps to be generated in Python. The census data contains demographic data on the state's population as well.

However, to use this US government data as initial data for the district generation in the GerryChain library some cleaning and proration is required. Fortunately, this has already been performed by the creators of GerryChain for all fifty states, In fact, election results data and other starting districting plans have been added, which allow for additional metric calculation and scenario simulations. For this analysis the cleaned data files for the state of Pennsylvania were used to generate the additional districts required. Pennsylvania was selected because it has a relatively diverse and politically balanced electorate in that it has two large urban areas on either side of the state (Philadelphia in the East and Pittsburgh in the West) and large rural areas in between the cities. Additionally, the state is landlocked geographically which makes the process of the MCMC image generation simpler, as we do not have to deal with islands and how to connect them to the network.

Generated Districts. Using the GerryChain library and the data for Pennsylvania, we generated a total of 27,900 images of realistic congressional districts, as well as their accompanying demographic data. Examples of the individual districts generated can be found in Figure 1. The MCMC function also has the ability to track important demographics, election results, and metrics throughout the Markov chain via, what it calls, *updaters*. The specific constraints and updaters that we used to generate a majority of our data can be found in Table 2.

We provide some remarks in relation to the constraints appearing in Table 2. For the population constraint, federal congressional districts based on the 2010 census averaged around 700,000 people per district across the country. This constraint was placed to

Table 2: Data Generation Constraints.

Type	Name	Value(s) / Description
Constraint	Population	Within 2% of 704,000 (initial average)
Constraint	Compactness Upper Bound	4 times the total cut edges in original plan
Initial Plan	Initial Districting Plan	2011 Congressional Map, 2018 Congressional Map, 538 Non-partisan Map
Updater	Black Population	The total African American population in the district
Updater	Non-black Population	The total non-African American population in the district
Updater	Area	The area of the district

ensure that this average was maintained when new districts were generated in the MCMC. The other compactness constraint was used to safeguard the generated districts from being unrealistically "snake-like", i.e., thin and uncondensed. As for the initial districting plan, several possible plans were used to test the impact on the data generated. The first two plans were actual congressional maps used in Pennsylvania from 2011-2017 and 2018-2022, and the third was a proposed non-partisan map by statistics website FiveThirtyEight[2].

The most important updaters to note were total population, total non-black population, total black population, and area. These four made up the features of the demographic data used in the final models. What we hoped to capture by including these features in the model are the racial undertones of gerrymandering and the urban vs. rural impact on gerrymandering. Race is one of the main demographics used to gerrymander as policymakers use it as a proxy for party preference and to disenfranchise African American voters, so it is important to capture that via features in our model. The urban vs. rural impact has been well documented in Borodin et al. [2018], and area of a district will be used as a proxy for this phenomenon. The guiding intuition being that given similar total district populations (as required by US federal law), larger districts with more area have a less dense population and are, therefore, more rural, and smaller districts are denser and more urban. Including these features in the models captured these effects.

Once the district images and their associated demographic data were generated, additional image preprocessing was required before use in the model. Many processing techniques and augmentations were attempted, but the final input images were zoomed in 30%, set to a sample mean of zero, and then normalized with the image's standard deviation.

A final important decision to note in the data generation phase is that districts were generated in three different colors – red, blue, and green. This is relevant because we wanted to examine the impact input image color had on the CNN that was developed.

[2]See https://fivethirtyeight.com/. The original 2011 map, based on the 2010 census, was challenged in the Pennsylvania Supreme Court and, ultimately, deemed gerrymandered. In response, the state Supreme Court redrew the districts in 2018 [11]. These were in place until the new districting plan, based on the 2020 census, took effect this year.

2.2 Gerrymandering Metrics

To detect gerrymandering, one first has to define what actually constitutes gerrymandering; this is in itself a non-trivial task. The literature has proposed several different metrics for deciding the extent to which a districting plan is gerrymandered. These can be broadly placed in two categories, *partisan* (related to vote shares) and *compactness* (related to district geometry) metrics (e.g., see [3, 12, 31]). Although several different metrics were analyzed and tested, the primary metric utilized in our work was the *efficiency gap score (EGS)* [31]. The EGS is a partisan metric because it relies on actual statewide election vote counts to calculate its score. The metric measures the relative difference between the parties' respective *wasted* votes [31], where wasted votes are votes acquired over the 50% winning threshold, i.e., votes that the party "did not need" to win the district. Formally,

$$\frac{n_w(P_1) - n_w(P_2)}{n},$$

where $n_w(P_1)$ and $n_w(P_2)$ is the total number of wasted votes for parties P_1 and P_2 respectively and n is the total number of votes. The metric is two-sided and ranges from $[-1, 1]$, with positive values indicating a district more gerrymandered for Democrats and negative values indicating a district more gerrymandered for Republicans.

Given the EGS reliance on real elections and the inconsistent nature of politics, we averaged EGS values across nine different statewide elections for each generated district's final score. This safeguarded the metric against any abnormalities caused by specific election circumstances such as the increased turnout of presidential cycles. The nine elections used are listed below.

- Senate Elections 2010, 2012, 2016
- Governor Elections 2010, 2014
- Attorney General Elections 2012, 2016
- President Elections 2012, 2016

Due to the continuous nature of the EGS metric, the thresholds for what constituted gerrymandering were relative and carefully considered. Histograms and summary statistics were generated to get a sense of the metric and potential thresholds. An example of one such histogram for the EGS metric can be seen in Figure 2. It was discovered that the mean for the EGS typically fell near -0.08 and its standard deviation, around 0.2. It is important to reiterate here the two-sided nature of the metric, thus requiring that two thresholds were set. According to gerrymandering legal experts (see [31]), 1.5 standard deviations away from the EGS mean in either direction constitutes a district plan that "is gerrymandered to an unusual extent and [...] has an unusually large impact on the makeup of the House" [31]. Using this intuition, we decided to set the thresholds of what constituted a gerrymandered district at 1 standard deviation away from the mean in either direction.

2.2.1 Multi-class classification model. We also built a multi-class model that classified gerrymandering in the districts as low, medium, or high levels of gerrymandering. A similar approach was used to set the thresholds for these three categories:

- the low class from $[-0.5, 0.5]$,
- the medium class from $[\mu - 1 \cdot \sigma, -0.5] \cup [0.5, \mu + 1 \cdot \sigma]$,
- the high class from $[-1, \mu - 1 \cdot \sigma] \cup [\mu + 1 \cdot \sigma, 1]$,

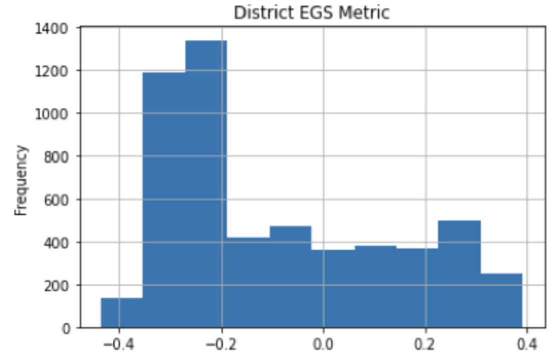

Figure 2: Histogram of the EGS Metric.

where μ is the mean of the distribution and σ is the standard deviation.

2.3 Machine Learning Model

Given that the data we generated consisted of numerical demographic data and district images, our model was required to handle this mixed input data. The key to building such models is to keep the inputs separate within their own deep learning architecture until they are of similar size and shape, then concatenate the outputs. Naturally, for the image data we used a convolutional neural network (CNN) and for the numerical data, a simple neural network.

For the CNN branch of the model we employed transfer learning with the VGG16 network[29]. An award winning ImageNet Challenge 2014 submission, the VGG16 network was an ideal choice for our analysis due to its elegance, accuracy, and overall simplicity. The network takes an input image and alternates between sets of convolutional layers and max pooling layers until the outputs are of the correct shape for use in fully-connected layers. The 16 in the VGG16 comes from the total number of hidden convolutional layers – there is a VGG19 version that is 3 layers deeper.

After the image data passed through the VGG16 network, the output passed through a flattening layer and then several fully-connected dense layers with dropouts in between. This resulted in the final layer of this branch outputting 4 nodes to the concatenation layer where the CNN branch was to combine with the neural network branch output. The nodes in these fully-connected layers were gradually reduced from 4,096 to 4 so as to not force too much of bottleneck from one layer to the next and the dropout layers safeguarded the model against the over-fitting that was initially observed.

For the numerical data, a much simpler network was utilized. Due to the low dimensions of the data (only 4 features) we found that more shallow neural networks performed best – finalizing that branch of the model with only two dense layers.

Once the mixed input data passed through their respective networks, they were then concatenated. After combining, the output passed through one more dense layer before a final output layer. The only difference between the binary and the multiclass classification models are the number of nodes in that final output layer

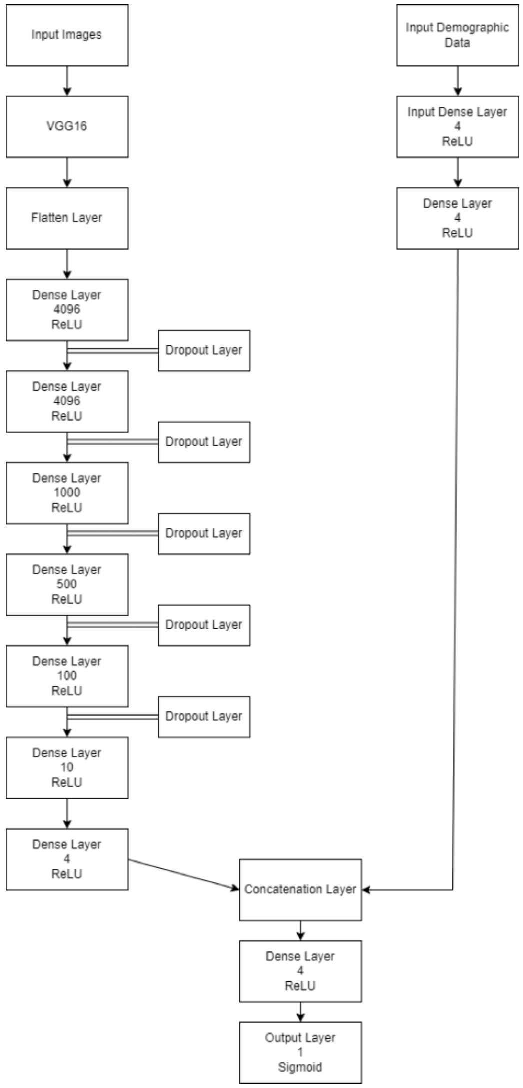

Figure 3: The general network model architecture.

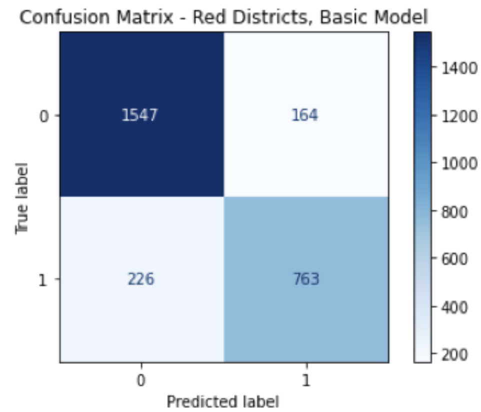

Figure 4: The confusion matrix for red-colored districts.

Table 3: The classification report for red-colored districts.

	Precision	Recall	F1-Score	Support
False	0.87	0.90	0.89	1711
True	0.82	0.77	0.80	989
Accuracy			0.86	2700
Macro avg	0.85	0.84	0.84	2700
Micro avg	0.85	0.86	0.85	2700

and the activation function (1 node and Sigmoid activation for binary, 3 nodes and Softmax activation for multiclass). A full visual representation of the model can be found in Figure 3.

Additional model parameters include a batch size of 32 where required, a learning rate of 0.001, and an epislon of 0.000005. In some cases, the learning rate and epislon were hypertuned with a search/tuning algorithm and those values were then altered, however, the best performing runs of the model were with those values. Further, the model was compiled with the Adam optimizer, which is widely accepted and known for its efficiency, low memory requirements, and versatility [19].

3 RESULTS AND EVALUATION

A model with high accuracy. Our model reached 86% test accuracy when run on 13,500 red-colored districts and their accompanying demographic data as the input data. The confusion matrix and classification report of the run that resulted in those numbers can be found in Figure 4 and Table 3.

Similar results were observed for the model runs trained with the other colored districts and it was determined that district color had no significant impact on the model. This proved that the CNN branch of the model was relying on the shape of the district more than the color – an indication that it was learning to detect the oddly-shaped trait that gerrymandered districts often have. Confusion matrices for runs with the blue and green district images can be found in Figure 5 and Figure 6 respectively. Their test accuracies reached 80% and 82%, respectively. Overall, reaching accuracies from 80% to 86% with the model offers compelling evidence for the use of machine learning models in gerrymandering detection and evaluation.

Initial Districting Plan. Another interesting question that was explored when generating additional data was the impact that the initial districting plan had on the MCMC algorithm. As described in the Data Generation section and noted in Table 2, three initial districting plans were used to initiate the Markov chains. According to [5], the starting plan should have no impact on the final data generation results after sufficiently many iterations. This proved to be true in our case as well – the data not only had similar metric histograms, but also resulted in comparable final testing accuracies.

Multiclass model. We also explored a multiclass model using the thresholds described previously. The three output classes were *low*, *medium*, and *high* levels of gerrymandering. Here our main finding was mostly negative, as the model did not perform nearly as well as in the case of the binary classification task. This is reflected in

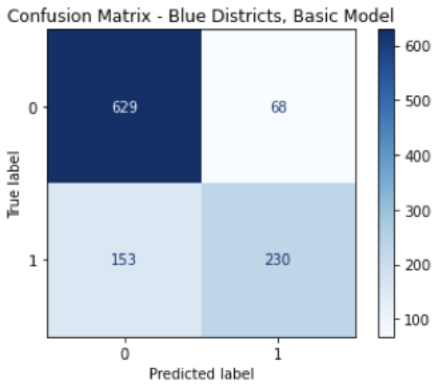

Figure 5: The confusion matrix for blue-colored districts.

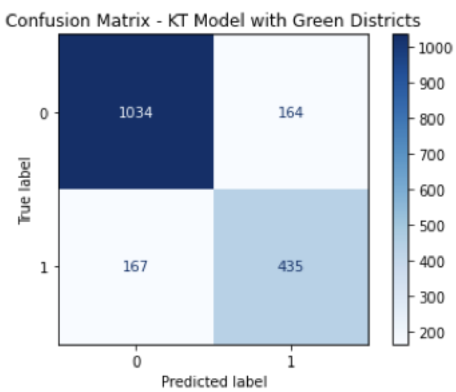

Figure 6: The confusion matrix for green-colored districts.

Table 4: The classification report for the multiclass model.

	Precision	Recall	F1-Score	Support
High	0.69	0.77	0.73	602
Medium	0.72	0.90	0.80	901
Low	0.00	0.00	0.00	279
Accuracy			0.71	1800
Macro avg	0.47	0.56	0.51	1800
Micro avg	0.59	0.71	0.64	1800

the lower accuracy of 71%, but more importantly in the fact that, as demonstrated in the confusion matrix and classification report, its behavior is essentially that of a (worse) binary classification model. The model only predicted two of the three classes and ignored the low class. This indicated to us that the data generated lends itself mostly to a binary classification, not a multiclass one, or perhaps more research is needed into identifying the correct thresholds for setting the ranges of the different classes. The results for this model are shown in Figure 7 and Table 4.

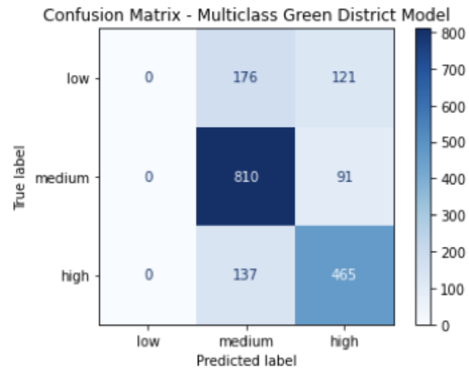

Figure 7: The confusion matrix for the multiclass model.

4 CONCLUSION AND FUTURE WORK

To the best of our knowledge, our work demonstrates the first compelling evidence that deep learning can be used as a tool for detecting gerrymandering in districting plans. Our trained models reached an accuracy of 86% with regard to the widely accepted EGS metric of Stephanopoulos and McGhee [31] (see also [5]); thus proving the concept of a predictive, machine learning model that can detect gerrymandering in individual districts. The models are clearly capable of learning the differences in districts, but they are also clearly dependent on the labeling metrics – that is the crux of the issue and where the bias lies. However, if a state's policymakers were to at least come together and agree upon a metric or set of metrics to quantify gerrymandering, and they ensured that their selection was based in statistics and in line with the established literature, then a model such as the one presented in this paper could be used as a standard for both detecting and measuring gerrymandering.

A reasonable question to ask in response to this conclusion is if a state agrees upon a metric that defines gerrymandering, why take it a step further and build a model? Why not just use the metric threshold(s) that were agreed upon to determine if a district is gerrymandered? However, that is what was done in the past and it was, ultimately, dismissed by the U.S. Supreme Court as too imprecise. A model, such as the one we built, that incorporates those metrics, district images, and district demographics makes a more compelling standard for gerrymandering detection and classification. If continued to be built upon, these models can prevent the dangers of gerrymandering in the U.S. and provide an unbiased, precise tool for policymakers to use.

Examples of areas to explore as future work include adding more demographic data to the neural network-branch of the model – especially voter registration data. When a U.S. citizen registers to vote in most states, they often register with a party as well (Republican, Democrat, etc.), this data would be useful in our model as another feature or two with the total number of registered Republicans/Democrats in the district. Additionally, there may be other demographics out there that could be incorporated into the model and examining feature importance metrics within the model (e.g., Shapley values [24]) may inform where to look for those additional features. Another clear direction is to further optimize the model architecture as well, testing additional networks architectures.

We are well aware that a majority of the bias in the model lies within the gerrymandering metrics and the data labeling process they are used for. Our choice of metric is based on the established literature on the topic [4, 31], as well as extensive research and time spent with these metrics. Still, further work could be done to examine other metrics or even develop new ones for use in a machine learning model. Another option regarding the labeling of the data is to explore possibilities for unsupervised learning. If one could just circumvent the biased step in the development process altogether and have a machine learning algorithm find its own labels in the data, that may be an effective course of action.

REFERENCES

[1] [n.d.]. Census Bureau Data. https://data.census.gov/cedsci/

[2] 2019. Rucho v. Common Cause. https://www.oyez.org/cases/2018/18-422

[3] 2022. GerryChain. https://github.com/mggg/GerryChain original-date: 2018-06-09T02:22:44Z.

[4] Sachet Bangia, Bridget Dou, Sophie Guo, Jonathan C. Mattingly, and Christy Vaughn Graves. 2015. Quantifying Gerrymandering : Data+ 2015 @ Duke. https://services.math.duke.edu/projects/gerrymandering/index-dataPlus.html

[5] Sachet Bangia, Christy Vaughn Graves, Gregory Herschlag, Han Sung Kang, Justin Luo, Jonathan C. Mattingly, and Robert Ravier. 2017. Redistricting: Drawing the Line. http://arxiv.org/abs/1704.03360 arXiv:1704.03360 [stat].

[6] Matthias Bentert, Tomohiro Koana, and Rolf Niedermeier. 2023. The complexity of gerrymandering over graphs: paths and trees. *Discrete Applied Mathematics* 324 (2023), 103–112.

[7] Allan Borodin, Omer Lev, Nisarg Shah, and Tyrone Strangway. 2018. Big City vs. the Great Outdoors: Voter Distribution and How It Affects Gerrymandering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 98–104. https://doi.org/10.24963/ijcai.2018/14

[8] Brian Brubach, Aravind Srinivasan, and Shawn Zhao. 2020. Meddling metrics: the effects of measuring and constraining partisan gerrymandering on voter incentives. In *Proceedings of the 21st ACM Conference on Economics and Computation*. 815–833.

[9] Jowei Chen and David Cottrell. 2016. Evaluating partisan gains from Congressional gerrymandering: Using computer simulations to estimate the effect of gerrymandering in the US House. *Electoral Studies* 44 (2016), 329–340.

[10] David Cottrell. 2019. Using computer simulations to measure the effect of gerrymandering on electoral competition in the US congress. *Legislative Studies Quarterly* 44, 3 (2019), 487–514.

[11] John Finnerty. 2018. Supreme Court will get to pick new map for state's congressional districts. https://www.meadvilletribune.com/news/local_news/supreme-court-will-get-to-pick-new-map-for-states-congressional-districts/article_ce46acd2-1edc-5987-86c4-ffc1f1c097aa.html

[12] Zachary Fisher. 2017. Measuring Compactness. https://fisherzachary.github.io/public/r-output.html

[13] Nikhil Garg, Wes Gurnee, David Rothschild, and David Shmoys. 2022. Combatting Gerrymandering with Social Choice: the Design of Multi-member Districts. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 560–561.

[14] Elmer Cummings Griffith. 1907. *The rise and development of the gerrymander*. Scott, Foresman.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Gregory Herschlag, Robert Ravier, and Jonathan C. Mattingly. 2017. Evaluating Partisan Gerrymandering in Wisconsin. http://arxiv.org/abs/1709.01596 arXiv:1709.01596 [physics, stat].

[17] Samuel Issacharoff. 2002. Gerrymandering and Political Cartels. *Harvard Law Review* 116, 2 (Dec. 2002), 593. https://doi.org/10.2307/1342611 Publisher: Harvard Law Review Association.

[18] Takehiro Ito, Naoyuki Kamiyama, Yusuke Kobayashi, and Yoshio Okamoto. 2021. Algorithms for gerrymandering over graphs. *Theoretical Computer Science* 868 (2021), 30–45.

[19] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. http://arxiv.org/abs/1412.6980 arXiv:1412.6980 [cs].

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[21] Justin Levitt and Michael P McDonald. 2006. Taking the Re out of Redistricting: State Constitutional Provisions on Redistricting Timing. *Geo. LJ* 95 (2006), 1247.

[22] Yoad Lewenberg, Omer Lev, and Jeffrey S Rosenschein. 2017. Divide and Conquer: Using Geographic Manipulation to Win District-Based Elections. (2017), 9.

[23] Chaochao Lu and Xiaoou Tang. 2015. Surpassing human-level face verification performance on LFW with GaussianFace. In *Twenty-ninth AAAI conference on artificial intelligence*.

[24] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[25] Nolan McCarty, Keith T Poole, and Howard Rosenthal. 2009. Does gerrymandering cause polarization? *American Journal of Political Science* 53, 3 (2009), 666–680.

[26] Daniel D Polsby and Robert D Popper. 1991. The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale L. & Pol'y Rev.* 9 (1991), 301.

[27] Christian P. Robert. 2016. The Metropolis-Hastings algorithm. http://arxiv.org/abs/1504.01896 arXiv:1504.01896 [stat].

[28] Peter H Schuck. 1987. The thickest thicket: Partisan gerrymandering and judicial regulation of politics. *Columbia Law Review* 87, 7 (1987), 1325–1384.

[29] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/abs/1409.1556 arXiv:1409.1556 [cs].

[30] Robin Smith. 2019. Duke Mathematics Has Its Day in Court. https://today.duke.edu/2019/03/duke-mathematics-has-its-day-court

[31] Nicholas O Stephanopoulos and Eric M McGhee. 2015. Partisan gerrymandering and the efficiency gap. *U. Chi. L. Rev.* 82 (2015), 831.

[32] William Vickrey. 1961. On the prevention of gerrymandering. *Political Science Quarterly* 76, 1 (1961), 105–110.

[33] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. 2018. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience* 2018 (2018).