Machine Learning Prediction of Political Orientation using Portraits of US Congress Members

JONAH EDMUNDSON* and MADISON GREENOUGH*, University of British Columbia Okanagan, Canada

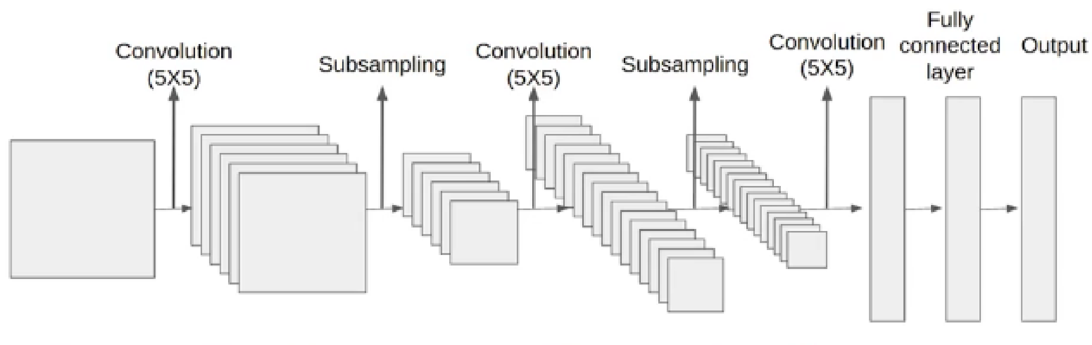

Fig. 1. LeNet-5 Architecture.

ABSTRACT GOES HERE!!!!

Additional Key Words and Phrases: datasets, neural networks, web-scraping, politics, facial analysis, CNN, classification

ACM Reference Format:
Jonah Edmundson and Madison Greenough. 2023. Machine Learning Prediction of Political Orientation using Portraits of US Congress Members. 1, 1 (April 2023), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

text

INSPIRATION: https://www.nature.com/articles/s41598-020-79310-1

*Both authors contributed equally to this research.

Authors' address: Jonah Edmundson; Madison Greenough, University of British Columbia Okanagan, Kelowna, BC, Canada.

2 METHODS

2.1 Dataset

The dataset for this project was obtained from the US Congress Members archive. Photos of each member as well as their party affiliation (republican or democrat) were recorded. Members with an affiliation of "independent" were removed. After data collection, the rows were shuffled, and the dataset was split into training (80%), validation (5%) and testing (15%). Images were also resized to 120 by 100 pixels (height by width) and converted to grayscale. Before CNN training, images were sequentially augmented thrice using the IMAGENET AutoAugment policy.

Fig. 2. Left: A subset of one batch before augmentation (64/128). Right: A subset of one batch after image augmentation (64/128).

2.2 Architecture

Three different architectures were trained and tested for classifying the images. These were LeNet-5, DenseNet, and a custom architecture (modified LeNet). The architectures of these NNs can be seen in tables 1 to 3. The DenseNet architecture can be further investigated at this GitHub repo.

2.3 Tuning

Many tuning attempts were made in order to raise the classification rate for the test data. Learning rate, batch size, number of epochs, dropout rate, number of features, weight decay and depth (DenseNet only) were systematically altered. The best values for each hyperparameter were discovered for each of the models, and can be seen in table 4.

Table 1

LeNet-5		
Features		
Convolutional Layer		Kernel Size = 4
Tanh Activation		
Max Pooling Layer		Kernel Size = 2
Convolutional Layer		Kernel Size = 14
Tanh Activation		
Max Pooling Layer		Kernel Size = 4
Classifier		
Linear	1408 In	120 Out
Tanh Activation		
Dropout		0.7
Linear	120 In	84 Out
Tanh Activation		
Dropout		0.8
Linear	84 In	2 Out

Table 2

DenseNet		
Features		
Dense Block		
Transition Block		
Dense Block		
Transition Block		
Dense Block		
Batch Normalization		
Classifier		
ReLU Activation		
Linear	24 In	2 Out

3 RESULTS

The LeNet-5 model outperformed DenseNet and our custom LeNet by a significant margin, even after extensive tuning (fig. 6). LeNet-5 obtained a test classification rate of 60.21%. DenseNet and custom LeNet achieved classification rates of 51.76% and 51.51.41%, respectively. These last two models perform only slightly better than random chance, while basic LeNet-5 performs decently well, but not amazing well. The convolutional filters of both convolutional layers for LeNet-5 can be seen in fig. 3.

Interestingly, figs. 4 and 5 show that while all the models were able to converge to a solution on the training data, this convergence did not lead to better classification for the validation set. For the training data, all three models were able to converge to the minimum cost in only two epochs. In contrast, for the validation data, the models just seem to jump around without improving for all 15 epochs.

Table 3

Modified LeNet-5 (Custom)		
Features		
Convolutional Layer		Kernel Size = 3
Sigmoid Activation		
Max Pooling Layer		Kernel Size = 2
Convolutional Layer		Kernel Size = 5
Sigmoid Activation		
Max Pooling Layer		Kernel Size = 2
Convolutional Layer		Kernel Size = 8
Sigmoid Activation		
Max Pooling Layer		Kernel Size = 2
Classifier		
Linear	1120 In	120 Out
Tanh Activation		
Dropout		0.8
Linear	120 In	84 Out
Tanh Activation		
Dropout		0.6
Linear	84 In	2 Out

Table 4

Hyperparameters	LeNet-5	DenseNet	Custom
Learning Rate	0.001	0.0012	0.0015
Batch Size	128	128	128
Epochs	12	15	15
Drop out	0.7, 0.8	0	0.8, 0.6
Features	12,000	12,000	12,000
Classes	2	2	2
Weight Decay	0.0002	0.0001	0.0001
Depth	*NA*	4	*NA*

4 DISCUSSION

Mention why test class. rate is higher than train (bc train data is augmented/all messed up, testing data is not, so it's easier to classify the test images)

The poor classification rate of our NN may be to any of the following:

(1) There is no actual difference in facial features between democrats and republicans.
(2) There are facial differences between democrats and republicans, but our dataset is too small to pick up on these subtle facial differences.
(3) The classifier section of the NN works well, but the convolution section is not properly picking up the meaningful facial features that are able to distinguish between the two parties.

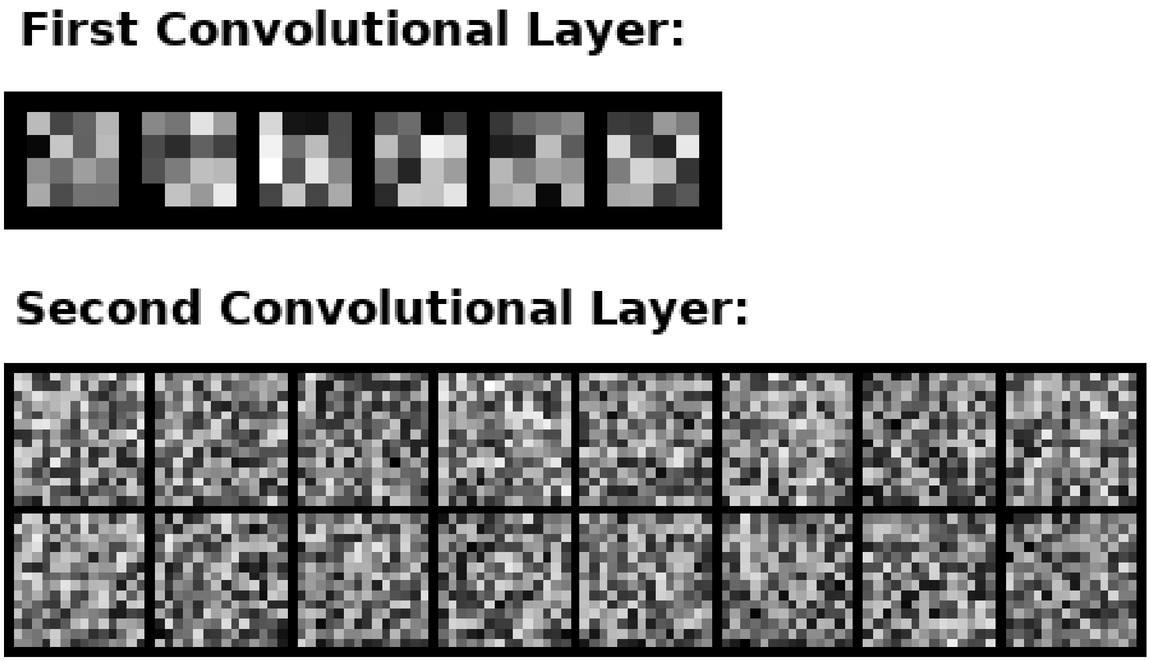

Fig. 3. The convolutional filters of the first and second convolutional layers.

(4) The convolution section of our NN is able to pick up on some meaningful signals, but the classification section is not performing well.
(5) The NN is only picking up features in the background of the image (and not the facial features), which are not predictive.

I think Number 5 is most likely.

5 LIMITATIONS & FUTURE WORK

Assumptions of this analysis due to the method of dataset construction are as follows:

- "Democrats" and "republicans" are good proxies for "liberal" and "conservative", respectively.
- Members of US congress are representative of democrats and republicans in general.
- Democrats and republicans have different facial attributes.

In order to rectify the low classification rate of our NN, as well as to iron out the assumptions listed above, a new dataset should be collected. Ideally, the dataset would be collected from public social media profiles such as Facebook or dating websites. This new dataset should also be much larger, ideally by several orders of magnitude. Once this dataset is collected, more complex NN architectures could also be attempted. If this were to be done, we speculate that the classification rate would increase, as was seen in the Kosinski (2021) paper.

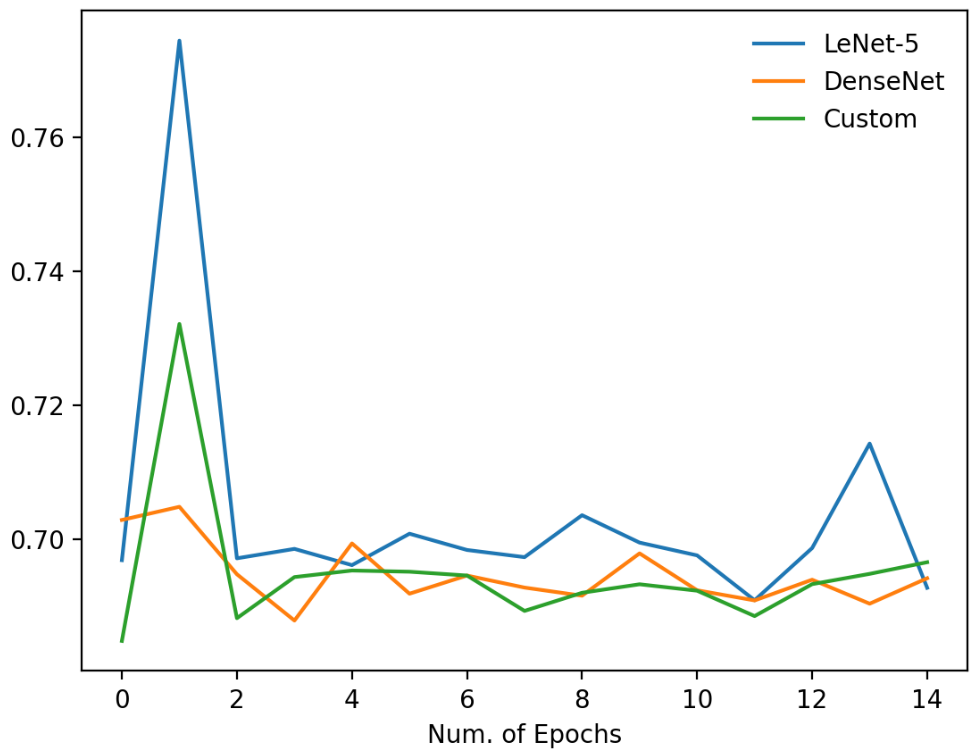

Fig. 4. The training cost over 15 epochs for the 3 architectures considered.

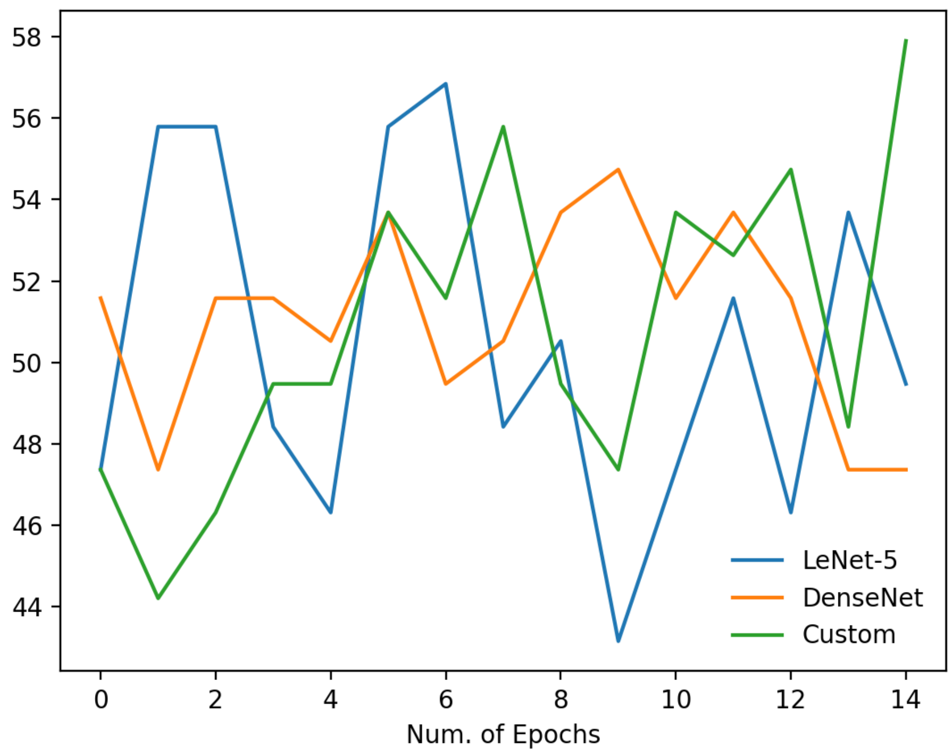

Fig. 5. The validation error over 15 epochs for the 3 architectures considered.

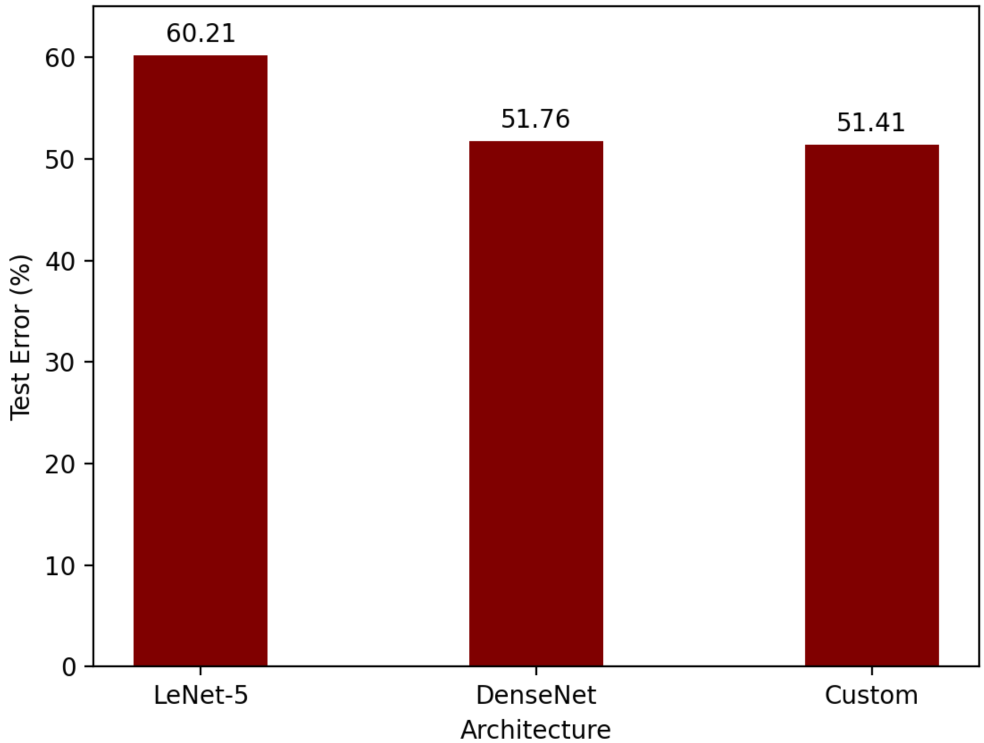

Fig. 6. The final testing error for the 3 architectures considered.