Political image analysis with deep neural networks

L. Jason Anastasopoulos* Dhruvil Badani† Crystal Lee‡
Shiry Ginosar§ Jake Ryland Williams¶

October 27, 2017

Abstract

Politicians and political organizations routinely interact with voters and the public at large using images, yet until recently, computational limitations have precluded efforts to gain systematic knowledge about how images function as a medium of political communication. New developments in machine learning, however, are bringing the systematic study of images within reach. In this paper, we provide a framework for political image analysis with deep neural networks, introduce neural networks and deep learning methods and discuss the promise and pitfalls of these techniques for political image analysis. Using a database of 296,460 photos from the Facebook pages of members of the U.S. House and Senate, we provide two illustrative exam-

*Corresponding Author; Microsoft Visiting Professor, Center for Information Technology Policy, Princeton University, Princeton, NJ, USA; Assistant Professor, Department of Public Administration and Policy, Department of Political Science, University of Georgia. Email: ljanastas@princeton.edu; Tel: +001 973.641.8258

†Department of Computer Science, UC Berkeley. Email: dhruvilb@berkeley.edu.

‡Pinterest.com. Email: isealya@gmail.com.

§Department of Computer Science, UC Berkeley. Email: shiry@eecs.berkeley.edu

¶Department of Information Science, Drexel University. Email: jakerylandwilliams@gmail.com.

ples of how these techniques can be used to study home style (Fenno 1978) in the digital age[1].

Keywords: neural networks; machine learning; deep learning; image analysis; home style; social media.

Word count: 9,947

[1]These data and all relevant `R` and `Python` code will be made available on the *Harvard Dataverse*: https://dataverse.harvard.edu/

1 Introduction

With the explosion of social media and improvements in camera technology, text posts and images have become a routine means through which political elites interact with the public (Esterling, Lazer, and Neblo 2013; Barberá 2015; Bond and Messing 2015; Accetti and Wolkenstein 2017; Ryan 2017a). While studies exploring the political uses of images in the literature have typically focused on using images as experimental treatments in vignettes (Hopkins 2015; Sen and Wasow 2016), exploring how exposure to images affects attitudes in natural experiments (Hainmueller and Hangartner 2013) or using images to assess the effects of candidate appearance on outcomes related to electability (Lawson et al. 2010; Hayes, Lawless, and Baitinger 2014; Horiuchi, Komatsu, and Nakaya 2012; Mattes and Milazzo 2014; Tingley 2014), there is little empirically based, systematic knowledge about how images function as a medium of political communication due to the difficulties of systematically extracting politically relevant image features at a large–scale.

Neural network techniques, described in detail below, give us the unprecedented ability to empirically study how images are used by political actors to shape public opinion. In this paper, we introduce neural network techniques and describe how these techniques can be used to identify and extract politically relevant information from images. Using a database of 296,460 photos from the Facebook pages of members of the U.S. House and Senate, we also

provide step–by–step examples of how these techniques can be used to study home style (Fenno 1978) in the digital age

2 Political images and image features

Before discussing neural network techniques in more detail, we first consider what a *political image* is and what constitutes a *politically relevant feature* of that image. Fortunately, we are able to stand on the shoulders of giants in the political science literature who provide much needed guidance for this task. A brief exploration of the literature on the political uses of images suggests that what distinguishes a "political image" from other types of images is that political images are created with the intention of persuading viewers to side with or against a political candidate, party, cause or organization (Hutchings, Walton Jr, and Benjamin 2010; Dilliplane, Goldman, and Mutz 2013; Prior 2013; Baker 2015).

In the case of political organizations the ultimate ends of the political images that they create are to further the interests of that organization (Wilson 1974; Hirsch 2016). Activities connected with these ends are multifaceted and may include membership recruitment, fundraising and other activities which accomplish these goals (Herrnson 1992; Wamsley and Zald 1973). If we assume that politicians' goals are also to further their own interests, the electoral connection (Mayhew 1974) suggests that political images function as tools which further politicians' re–election goals (Giger and Klüver 2016;

Lopes da Fonseca 2017; Ryan 2017b). Under this premise, images are used by politicians to increase their likelihood of re-election or otherwise advance their careers.

If political organizations and politicians utilize images as a means of furthering their interests, the primary goal of political image analysis, then, is to understand *how* images are used to further their interests. To that end, we first discuss how objects and people, features of political images which can be identified using neural networks, can accomplish these ends and provide two illustrative examples demonstrating how object identification and racial classification with neural networks can be used to understand home styles (Fenno 1978) in the digital age. These are by no means an exhaustive list of the politically relevant aspects of images, but rather are meant to provide a starting point toward the identification of image features which can shed light on political communication through visual media.

Figure 1 – Former Texas Governor Rick Perry posing with rifles in a gun shop. *Source* http://www.politico.com/gallery/politicians-with-guns

2.1 Objects

Human beings recognize and use objects to convey symbolic meanings in childhood as early as 18 months (Tomasello, Striano, and Rochat 1999). Indeed, the use of objects as signifiers of meaning in everyday life is so common that it generally goes unnoticed. Flags are objects which represent nation–states, trees are objects used to represent environmentalism and so on. In the political context, objects may take on other abstract meanings which, in the case of political figures, relate to their personal qualities or policy positions or, in the case of nation–states and other forms of political organization, may relate to shared values and mores.

The American flag, for example, an object and a symbol, represents the geopolitical construct of the United States, but depending on context can represent American values and ideals, American patriotism, American military power or a combination of all three. Objects such as firearms and military equipment can be used to represent specific policy stances such as support or opposition to gun control, support for veterans, military interventions and so on. Thus, a politician might convey opposition to gun control and/or support for the Second Amendment by posing in a gun shop as former Republican Texas Governor Rick Perry does in Figure 1.

2.2 People

People in political images are politically relevant to the extent that their inclusion in a political context can convey information about a political figure, organization or institution. This can be accomplished through the inclusion in an image of well-known person (celebrity, political figure, etc.) or unknown people in different contexts. When posing with unknown individuals, visible features (race, gender, veteran status, age etc.) of these individuals can shape how a political figure or organization is perceived.

Figure 2 – A photo collage posted on Rep. Louise Slaughter's (D-NY 25) Facebook profile where she is posing with African-American, Hispanic and white constituents at a festival in her district. *Source:*https://www.facebook.com/RepLouiseSlaughter/.

Political figures themselves can use the visible qualities of the individuals that they pose with in a similar manner. A politician seeking to increase turnout among members of certain constituency groups, for example, may

Figure 3 – House Speaker Paul Ryan Poses with a Vietnam Veteran. *Source:* https://www.facebook.com/paulryanwi/.

consistently pose with the group(s) that they are targeting. Representative Louise Slaughter (D–NY 25) in Figure 2, for example, poses with African–American and Hispanic members of her constituency at a festival in her district while House Speaker Paul Ryan in Figure 3 poses with a veteran.

Regarding famous individuals, the information conveyed depends largely upon how the famous individual is perceived by the viewer. For a recent example, Figure 4 contains a photo of Republican New Jersey governor Chris Christie embracing President Obama after a visit to assess some of the damage caused by Hurricane Sandy in 2013. While right-leaning media sources such as Breitbart excoriated Christie, other news sources praised Christie's bi-partisanship.

Figure 4 – New Jersey governor Chris Christie embraces President Obama during his visit to the state after Hurricane Sandy in 2013. *Source* http://www.breitbart.com/big-government/2015/07/01/inside-hug-gate-the-online-meme-that-chris-christie-cant-shake/

3 Political image analysis with neural networks

Neural networks provide the key to extracting politically relevant features from objects and people in images. In our illustrative examples below, we apply these techniques to a database of 296,460 photos collected from the Facebook pages of members of the House and Senate to presentation of self in the home styles (Fenno 1978) of members of Congress. Before delving into these analyses, however, we first provide an overview of neural networks and deep learning methods.

3.1 Introduction to neural networks

Artificial neural networks are a machine learning technique originally inspired by neurological processes (Hebb 1949; McCulloch and Pitts 1943; Rosenblatt 1958). If we think of the brain as an information processing system which absorbs information from the environment via the senses and uses this information to create models which make sense of the external world, the smallest unit of this system which does all of the information processing are the neurons, or brain cells, which are essentially a series of parallel networks.

Figure 5 – Insert image of neuron here

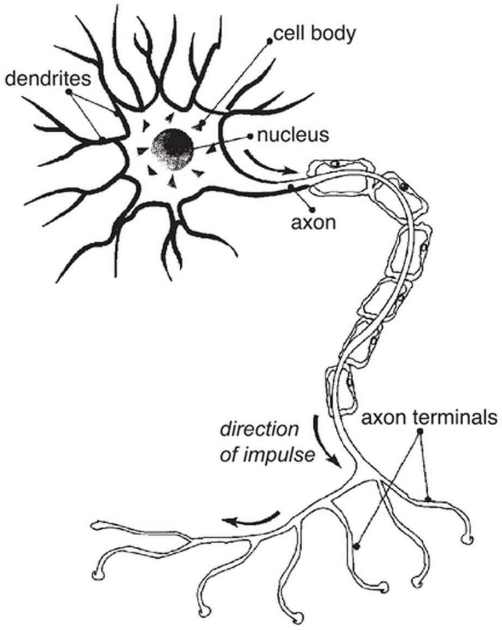

Figure 6 – Depiction of a neuron in the brain. Source http://blanco.io/assets/images/neural-nets/

Neurons in the brain contain three components: (1) *dendrites* – which

take in information as signals; (2) *cell body* – which process the signal information and; (3) *axon* – which transfers the processed information to other neurons. Figure 5 depicts a typical neuron in the brain. In the same way that neural networks in the brain take in information that is passed to the senses from the external environment and spreads that information to other neurons, artificial neural networks pass data through the *neuron*, the basic unit of artificial neural networks, which contains an *activation function* that "processes" the data.

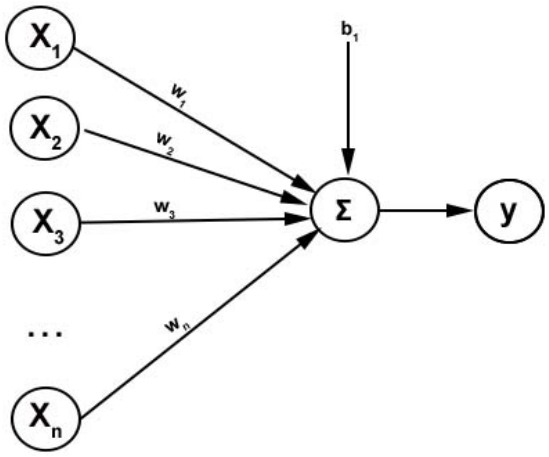

Figure 7 – Perceptron model with n inputs and no hidden layers. Inputs (X_i are variables, w_i are weights that the variables are multiplied by. The neuron Σ sums the variables multiplied by the weights, adds a bias b_1 and predicts an outcome Y using a threshold activation function.

We first describe the simplest neural network model called the *perceptron* (Rosenblatt 1958) and provide some mathematical intuition for its components. We then discuss how more complex neural network models

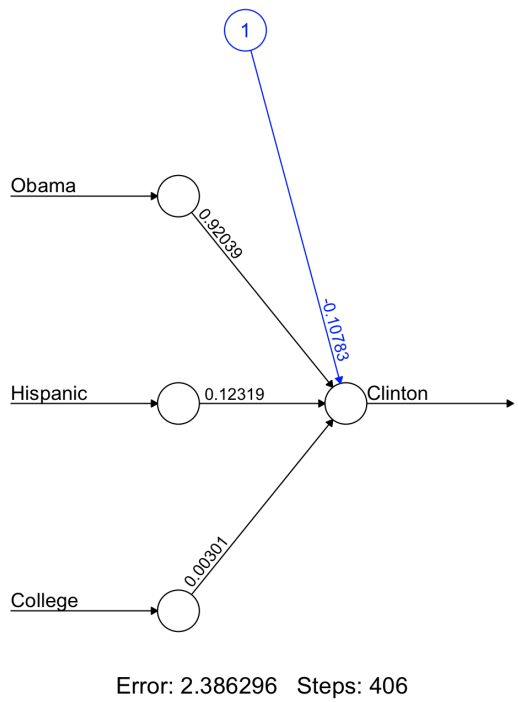

Figure 8 – Estimated perceptron model with three inputs and no hidden layers. The outcome *Clinton* equals 1 if Hillary Clinton's 2016 county level vote share was $\geq 50\%$ and 0 otherwise. Inputs predicting this are: (1) Obama's vote share in 2012 (*Obama*); (2) the % hispanic (*Hispanic*) and; (3) the % of college educated people (*College*). *Error* is the final mean squared error (MSE) of the model, *Steps* are the number of epochs requires to estimate weights which minimize the MSE.

can be built using this framework. The neural network model in Figure 7 has a number of components. The *inputs* $X_1, X_2, \cdots, X_n$ in the traditional statistics sense, are *variables*. The inputs are multiplied by a set of *weights* $w_1, w_2, \cdots, w_n$ and then passed through the *neuron* Σ in the form of the following linear equation.

In Figure 8 we estimated a perceptron model of whether Hillary Clinton had a majority vote share in a county in 2016 (*Clinton*) using the variables:

$$\Sigma = \tau = b + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \tag{1}$$

Obama – Obama's 2012 county level vote share; *Hispanic* – the county % hispanic and; *College* – the % college education.

A bias term b, or constant is usually added and the value of τ is used in estimation. After τ is computed an *activation function* f, serves as a means of predicting the *output* y of the observations of that variable. For the purposes of this illustration, the output of these observations is a binary *class label* or categorical variable, but the output can also take on continuous or discrete numerical values.

$$f(\Sigma) = y = \begin{cases} 0 & \tau < \tau^* \\ 1 & \tau \geq \tau^* \end{cases}$$

In the perceptron model, the activation function shown above, determines the *class label* or category that the outcome y is classified as based on some threshold value τ^* which is determined by Equation 1. Equation 1 may be familiar as the linear model for ordinary least squares (OLS), but there are several key differences. First, and most importantly, the linear function in Equation 1 is not a model that we are estimating for the purpose of *statistical inference*. As a result, we are only concerned about estimating

the weights w in order to *maximize prediction success* and thus the typical properties of statistical estimators that we are generally concerned about such as biasedness are important only to the extent that they impede our ability to to predict the outcome y.

Apropos of this, *learning* the weights which maximize the predictive success of the neuron are accomplished with the *backpropagation* algorithm, described in more detail below. Before discussing this, however, we elaborate further on two aspects of the neural network: (1) the activation function; (2) building a neural network with multiple neurons and hidden layers.

3.1.1 Activation functions

The non–linear activation functions which the linear data in Equation 1 are passed through are an important part of what give neural networks their power to estimate complex non–linear functions from data. An activation function $\sigma(\cdot)$, takes the general form:

$$\sigma\left(\sum_{i=1}^{n} w_i X_i + b\right) \tag{2}$$

Where $\sigma(\cdot)$ can be smooth sigmoid (s–shaped) functions such as the logistic function or tangent function, or can be discontinuous like a step function. Activation functions which give good classification performance and are commonly used include the logistic function and its normalized version the *softmax* function, the *tanh* function and the rectified linear unit (ReLU)

Activation Function	**Equation**
Softmax	$logit(\tau) = \frac{exp(\tau)}{\sum_{j=1}^{N} exp(\tau)}$
tanh	$tanh(\tau) = \frac{1 + exp(-\tau)}{1 + exp(-\tau)}$
ReLU	$relu(\tau) = max(0, \tau)$

Table 1 – Examples of activation functions

function.

Table 1 provides some examples of common activation functions used for classification purposes. In multiple neuron, layered neural networks, a combination of these, or other functions may be used depending on the problem. An unfortunate aspect of neural networks is that there are generally no clearly stated mathematical rules regarding which activation function or combination of functions will be best for any given problem. However, certain activation functions such as the *tanh* function, combined with data normalization techniques, will allow for more efficient estimation of weights through the backpropagation algorithm discussed in more detail below (LeCun et al. 2012).

3.1.2 Building a neural network: the hidden layer

The perceptron model discussed above may perform well for simple classification tasks, but will often fare poorly when classification problems become more complex. For example, identifying features from images from pixel in-

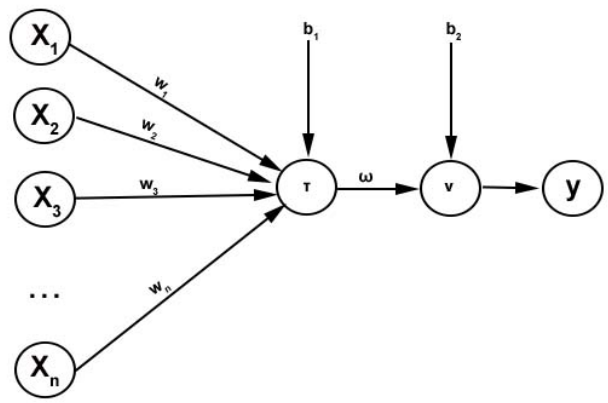

Figure 9 – One hidden layer model. Inputs X_i are variables, w_i are weights that the variables are multiplied by while ω_i are weights multiplied by the activation function and neuron 1 $\sigma(\tau) = \nu$. $\tau = b_1 + \sum_i w_i X_i$; $\nu = b_2 + \omega[\sigma(\tau)]$

tensities which the deep learning methods that we employ below accomplish, requires estimating a set of class labels representing objects (i.e. dogs, cats, cars, etc.) as a complex, and unknown, non–linear function of pixel intensity data. To accomplish this, and other more complicated tasks, we need to build neural networks that contain multiple neurons with multiple *hidden layers* that the original data are passed through in order to allow for the flexible estimation of a complex non–linear function.

From a mathematical perspective, the *hidden layer* uses the output from a previous neuron or a previous hidden layer as input for a new activation functions for which new weights are estimated. We illustrate how the hidden layer works using a simple one hidden layer neural network. Figure 9, is a model of a one hidden layer neural network that is estimated with election

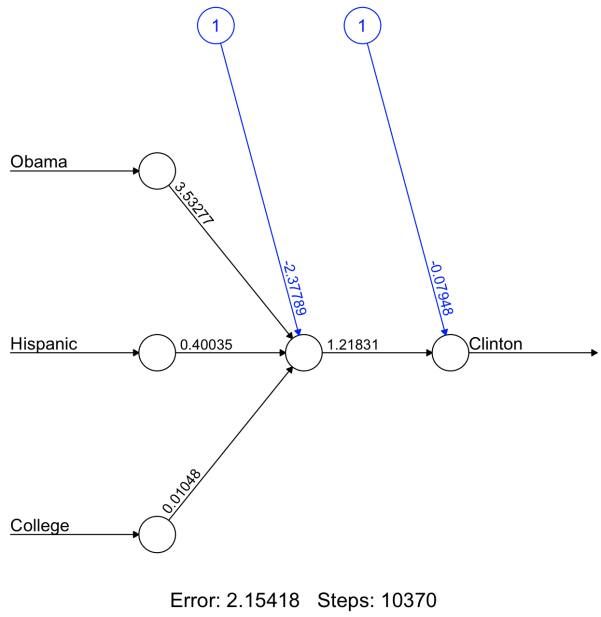

Figure 10 – One hidden layer model with three inputs. The outcome *Clinton* equals 1 if Hillary Clinton's 2016 county level vote share was $\geq 50\%$ and 0 otherwise. Inputs predicting this are: (1) Obama's vote share in 2012 (*Obama*); (2) the % hispanic (*Hispanic*) and; (3) the % of college educated people (*College*). *Error* is the final mean squared error (MSE) of the model, *Steps* are the number of epochs requires to estimate weights which minimize the MSE.

data in Figure 10 using the same variables as the perceptron model discussed above. Note that with one hidden layer, the final MSE decreases from about 2.39 from the model in Figure 8 to 2.15 but significantly more epochs (10,370) are required to estimate the more complicated model as compared to the perceptron (406) .

For the purposes of illustration we will assume that the activation function of the first neuron and second neurons are *logistic*.

If our input function is thus:

$$\tau = \sum_{i=1}^{N} w_i X_i + b_1 \tag{3}$$

Passage through the first activation in neuron 1 gives us:

$$y = \sigma(\tau) = \frac{1}{1 + exp(\tau)} = \frac{1}{1 + exp(\sum_{i=1}^{N} w_i X_i + b_1)} \tag{4}$$

The output from the equation $\sigma(\tau)$ is the passed to neuron 2, a new activation function $\phi(\dot{})$ is applied and new weights are added. Thus the second activation function is now:

$$\nu = \phi(\sigma(\tau)) = \omega_1 y + b_2 = \omega_1 \left(\frac{1}{1 + exp(\sum_{i=1}^{N} w_i X_i + b_1)} \right) + b_2 \tag{5}$$

$$z = \frac{1}{1 + exp(-\nu)} \tag{6}$$

Here, it becomes clear that while Equation 5 may appear to be linear in z it is in fact a more complicated non–linear function containing information from the original inputs. As inputs, layers and neurons are added, these functions become more and more complex, which cause two problems: (1) the probability of *overfitting* the model to the training data increase and, from a social science perspective, the interpretation of the weights become less and less clear. This is why neural networks are often criticized as "black boxes" that do not allow the researcher to understand which variables/inputs

Neuron 1	$\tau = \sum_{i=1}^{N} w_i X_i + b_1$
Activation 1	$y = \sigma(\tau) = \dfrac{1}{1 + exp(\tau)} = \dfrac{1}{1 + exp(\sum_{i=1}^{N} w_i X_i + b_1)}$
Hidden Layer (Neuron 2)	$\nu = \phi(\sigma(\tau)) = \omega_1 y + b_2 = \omega_1 \left(\dfrac{1}{1 + exp(\sum_{i=1}^{N} w_i X_i + b_1)} \right) + b_2$
Activation 2/Prediction	$z = \dfrac{1}{1 + exp(-\nu)}$

Table 2 – Representation of a two–layer neural network in equations

are contributing most to the the classification (Tomandl and Schober 2001).

3.1.3 Backpropagation and gradient descent: learning with neural networks

In the previous section, we discussed each of the main elements needed to build a complex neural network classifier including *inputs/variables*, *connections/weights* and *activation functions*. What we have omitted to this point, however, is the crucial feature of neural networks which allows them to learn from data: estimation of the weights, ω and w in the two neuron, single–hidden layer network above.

$$\epsilon = \frac{1}{N} \sum_{i} (y_i - \hat{y}_i)^2 \tag{7}$$

As mentioned above, neural networks in the context of political image analysis are a supervised algorithm used to *predict* new outcomes or class

labels from data. There are two keys to accomplishing this: (1) *building an architecture to maximize predictive success*: structuring the neural network by adding/subtracting neurons, hidden layers and choosing activation functions and; (2) *estimating weights in the hidden and input layers* to maximize predictive success. We will discuss building an architecture in more detail below, but here we turn our attention to weight estimation.

In a typical OLS regression problem with one variable, the model might appear very similar to the first neuron from Table 6:

$$\tau = \sum_{i=1}^{N} w_i X_i + b_1 + \epsilon \tag{8}$$

When we use OLS, as in the equation above, we might be interested in modeling an outcome τ as a linear function of X_i. To accomplish this, we would use the method of *least squares* to estimate w_i and b_1 such that the sum of squared errors are minimized:

$$\arg\min_{w_i, b_1} \sum_{1=1}^{N} (\tau - \sum_{i=1}^{N} w_i X_i - b_1)^2 \tag{9}$$

Equation 9 can be easily solved for w_i and b_i by taking the derivatives of the sum of squared errors (SSE) with respect to the constant and the weights. The goal of estimating weights in neural networks is identical to the goal of estimating weights in the context of OLS: finding weights which minimize the SSE. Unfortunately, however, in the case of neural networks, especially those with multiple neurons and multiple layers, analytical solutions are rarely

available. How then, can weights be estimated across all types of neural networks from the simplest to the most complex?

As it turns out, *backpropagation*, which employs the chain rule and *gradient descent* which provides a method for adjusting weights in the direction of decreasing error, can at least theoretically estimate global minimia for a wide variety of neural network architectures[2]. We describe the backpropagation algorithm and gradient descent in more detail below.

Define the *mean squared error* for our purposes as the average sum of squared errors between the class labels y_l and the values predicted by the neural network y_i as described in the two–layer, two–neuron network toy example discussed above. A neural network which best predicts the output value is the one in which the weights are estimated in all layers to minimize some *loss function* which, for this example, is the sum of squared errors, sometimes referred to as the L_2 loss function[3]

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N}(y_{li} - y_i)^2 \tag{10}$$

Given the loss function in Equation 10, our goal is then to derive weights $\mathbf{w} = \{w_1, \cdots, w_P, \omega_1, \cdots, \omega_P\}$ such that ϵ is minimized.

[2]We say *theoretically* here because it is still possible for these techniques to get stuck at local minima although several researchers have been working on techniques to avoid these issues.

[3]There are many different types of loss functions using for estimating neural network weights which serve different purposes. Some, such as the *hinge loss* result in faster convergence but produce noisier results, while others, such as the commonly used L_1 and L_2 produce more accurate results but are more computationally intensive (Vapnik and Vapnik 1998).

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{N} (y_{li} - y_i)^2 \tag{11}$$

Backpropagation and gradient descent allow us to do this by (1) finding out the direction in which we need to adjust $\mathbf{w}$ in order to minimize the loss function and; (2) incrementally adjusting $\mathbf{w}$ in that direction. The steps involved in this process are:

1. **Forward pass** – Weights are randomly initialized and the data, multiplied by weights, is passed through the network.

2. **Backpropagation of errors** – Error gradients with respect to the weights in each layer are calculated using the chain rule.

3. **Weight adjustment using gradient descent** – Weights are adjusted in the direction of the negative gradient for the next forward pass and until some convergence criteria is reached.

A full completion of each of these steps is called an *epoch*. Weight estimation thus often requires thousands of epochs before convergence criteria is reached. We illustrate steps 2 and 3 using the two neuron, one hidden layer model described above.

After the first forward pass, predicted values $z_{l,m}$ are generated from the model and the error ϵ is calculated. The backpropagation phase then uses the chain rule to calculate the gradient for the weights at each layer:

$$\frac{\partial \epsilon}{\partial \omega} = \frac{\partial \epsilon}{\partial z}\frac{\partial z}{\partial y}\frac{\partial \nu}{\partial y} \tag{12}$$

$$\frac{\partial \epsilon}{\partial w} = \frac{\partial \epsilon}{\partial \omega}\frac{\partial y}{\partial \tau}\frac{\partial \tau}{\partial w} = \left[\frac{\partial \epsilon}{\partial z}\frac{\partial z}{\partial y}\frac{\partial \nu}{\partial y}\right]\frac{\partial y}{\partial \tau}\frac{\partial \tau}{\partial w} \tag{13}$$

Equation 12 is the gradient of the errors with respect to the weights in the hidden layer and Equation 13 is the gradient of the errors with respect to the weights in the input layer. All of the sub–gradients in each of these equations can be solved regardless of how many hidden layers there are in a neural network. Backpropagation provides us with the ability to link the error ϵ to the weights in the nth layer via the chain rule.

After these gradients are calculated, the total gradient γ_m for the pass or epoch m is computed by summing the gradient over each of the observations for each weight:

$$\gamma_m = \sum_{n=1}^{N}\left[\frac{\partial \epsilon}{\partial w_m}\right]_n \tag{14}$$

The total gradient γ_m is then used to adjust the weights in the $m+1$ epoch:

$$w_{m+1} = w_m + \Delta w_m$$

$$\Delta w_m = -\lambda \gamma_m$$

The steps above are repeated over as many steps as it takes before some stopping criteria $-\lambda\gamma_m < \eta$ is reached. The parameter λ is referred to as the *learning rate* and determines how large of a step the weights are adjusted by. The learning rate can be a fixed value or can itself be randomly adjusted. While this is a topic of significant interest, it is beyond the scope of this paper.

4 Deep neural networks and image as data

The exposition of neural networks above provides the foundation for understanding more complex neural network techniques referred to as *deep neural networks* or *deep learning* techniques. These techniques are known as "deep" neural networks because they have many hidden layers and often contain hundreds of neurons and a multiplicity of activation functions. Despite this however, the structure and learning methods through which these networks are trained are fundamentally the same as those of the simple two neuron, one hidden–layer neural network. While there are many varieties of deep neural network techniques, we focus here on a specific type of deep learning technique known as convolutional neural networks (CNNs) because they have achieved the greatest success for a wide variety of image classification tasks and are tailored to the idiosyncrasies of image data (Krizhevsky, Sutskever, and Hinton 2012). Below we provide a brief discussion of convolutional neural networks and image as data before discussing our applications in more

detail.

4.1 Image as Data

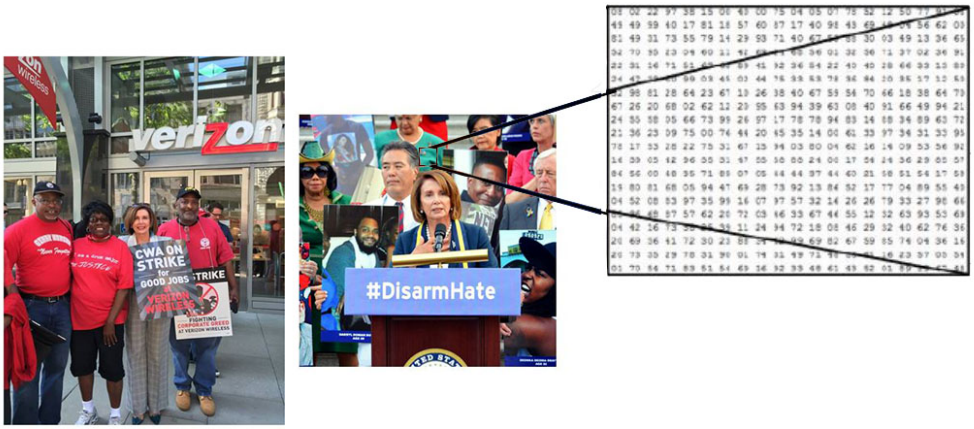

Figure 11 – Images of Nancy Pelosi as represented on a machine by a matrix of pixel intensities.

When we view an image on a computer screen, we are seeing a collection of pixels stacked in a certain order. Grayscale (black and white) images are represented on a machine as a single matrix of pixel intensity values. The most common format for pixel intensity values is the *byte image* which is stored as an 8-bit integer that takes on a range of integer values between 0 and 255. Color images are also stored as pixel intensity values, but instead of containing pixel values across a light/dark dimension, color images contain pixel intensity values across three dimensions, or channels: red, green

and blue. Pixel intensity values for both grayscale and color images are thus represented as one or three matrices of pixel intensity values in the range of 0 to 255, respectively. Because images, both color and grayscale, are represented on machines as either one or three matrices, each image in a machine is typically represented as a series of *tensors* or arrays of multidimensional arrays.

4.2 Convolutional Neural Networks

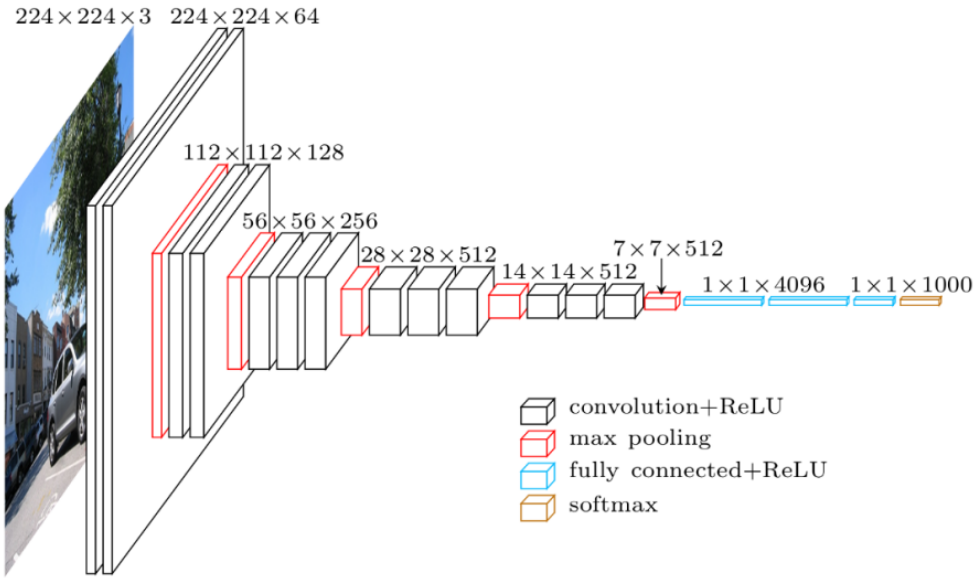

Figure 12 – VGG16 convolutional neural network architecture as implemented on *TensorFlow*[4].

A convolutional neural network (CNN) is a deep (multiple layered) neural network model which is tailored for image analysis. Ordinary neural networks run into significant computational inefficiencies when dealing with

image data. As mentioned above, image pixel intensities have three color channels (RGB) and are thus represented by a tensor of pixel intensity values[5]. Thus, for example, if each pixel intensity value is considered as an observation for a $50x50$ image, the input for one neuron would require estimating a vector of $50x50x3 = 7,500$ weights. This, unfortunately, does not scale well to even small images. A $250x250$ image, for example, would require estimating $250x250x3 = 187,500$ weights for only *one* neuron.

In order to handle image data more efficiently and effectively, CNNs use three different types of layers : (1) **convolutional layers** – which slide a series of *filters* across the image to produce a series of *feature maps* (described in detail below); (2) **pooling layers** – which perform downsampling or dimensionality reduction on the feature maps and; (3) **fully connected layers** – which use the downsampled image data from the pooling layers to produce image classes or labels. Figure 12 is an example of a high–performance CNN architecture used for image classification as implemented in the popular deep learning package *TensorFlow*. Each type of layer is described in more detail below.

4.2.1 Convolution

Convolutional neural networks work by first reducing images to a series of *feature maps* by sliding, or convolving a *filter* of a certain pixel size across the image and computing the dot product. Each element of the feature map

[5]In contrast, grayscale images are represented by only one matrix of pixel intensities.

Grayscale 5x5 image	$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 0 & 4 & 0 & 1 & 1 \\ 1 & 0 & 2 & 0 & 1 \end{bmatrix}$
Filter:	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$
Convolution over $x_{11}, x_{12}, x_{21}, x_{22}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = 1 * -1 + 0 * 0 + 0 * 0 + 1 * 1 = 0$
Feature map	$\begin{bmatrix} \mathbf{0} & 2 & 1 & 0 \\ 3 & 0 & -2 & -1 \\ 3 & -3 & 0 & 1 \\ 0 & -2 & 0 & 1 \end{bmatrix}$
ReLU feature map:	$\begin{bmatrix} 0 & 2 & 1 & 0 \\ 3 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 3 – A $5x5$ grayscale image represented as a matrix and a $2x2$ filter. When the upper left hand portion of the image captured by the $2x2$ matrix with 1's across the diagonal is convolved with the filter, the $2x2$ output is converted to a scalar. After sliding this filter across the image, it reduces the $5x5$ image to a $4x4$ feature map. The $ReLU$ activation function transforms each value of the feature map according to the following rule: $max(0, input)$.

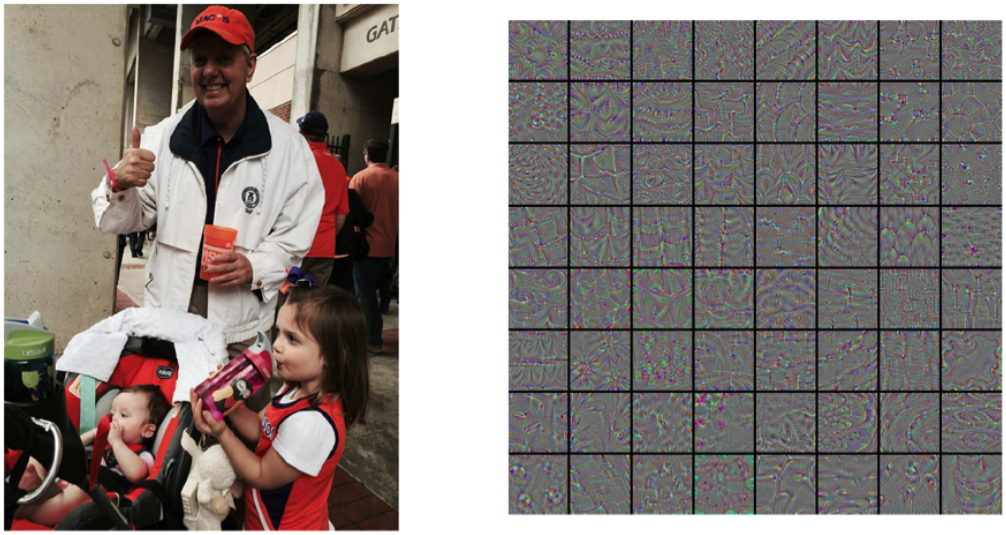

Figure 13 – Feature maps of one of Lindsey Graham's (R–SC) Facebook photos. These feature maps are one slice of the VGG16 convolutional neural network model shown in Figure 12 before being downsampled in the first max–pooling layer.

matrix is then transformed by the *ReLU* or *recified linear unit* activation function as discussed above according to $max(0, input)$. In practice, the numerical values of these filters are actually learned from the data, i.e. the values which comprise the filters are the equivalent of the learned "weights" discussed in the previous section, but the *number of filters*, their *size* and their *stride* or how far across the image they slide, are parameter choices that have to be made. Although this field is relatively new, software packages in *Python* and *R* such as *TensorFlow* contain pre-loaded models which have performed well in image classification tasks. Figure 13 contains feature maps of one of Lindsey Graham's (R–SC) Facebook photos. These feature maps are one slice of the VGG16 convolutional neural network model shown in

$$\begin{bmatrix} 0 & 2 & 1 & 0 \\ 3 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \max \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix} & \max \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\ \max \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} & \max \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 1 \\ 3 & 1 \end{bmatrix}$$

ReLU Feature map **Max–Pooling Operation**

Table 4 – Max pooling of the image feature map from Table 3

Figure 12 before being downsampled in the first max–pooling layer.

4.2.2 Pooling

Pooling is a procedure in which feature map data are downsampled by choosing a window size from the feature map matrix and selecting the largest or smallest value. In practice, selecting the largest value or "max pooling" has been shown to be the more effective means of dimensionality reduction. Table 4 demonstrates how max pooling works. The feature map is is divided into portions and only the maximum value of those portions are passed to the next layer.

4.2.3 Fully connected layer

These processes of convolution and pooling continue until the three dimensional image data are reduced to a single vector. It is at this point that an ordinary neural network is estimated, typically with *ReLU* and *softmax* activation functions. This stage of the process is known as the fully connected layer and can be seen in Figure 12 as the final stage prior to the class label

assignment. Below we provide two illustrative examples of how convolutional neural networks can be used to study political images with a focus on home style (Fenno 1978). In the first example, we demonstrate how a pre–trained object recognition model can help us understand more about object usage and imagery among partisans through the images that they post on Facebook and in the second example, we train a convolutional neural network classifier to identify the race of the people that members of Congress pose with in a study on partisan racial representation through images.

5 Visible Home Styles

Home styles are idiosyncratic behaviors that members of Congress adopt to gain trust among their constituents. Two components of home style which can be represented visually include Washington activities and presentation of self (Fenno 1978). *Visible home styles*, then, refer to the ways in which politically relevant image features, such as objects and people as discussed above, can convey either aspects of presentation of self or Washington activities. For example, Republican House members may use objects such as military equipment in their photos as a means of signifying support for the veterans and military families in their districts (empathy) or to demonstrate that they are bringing in federal subsidies (Washington activities) to their districts.

Fenno outlined four "concentric circles of constituency" when describing

how representatives view their constituencies: the "geographic constituency," which is comprised of all members of representative's geographic districts; the "re–election constituency" which includes people whom the representative believes are most likely to get her re–elected; the "primary constituency" which includes the representative's most reliable supporters and finally the "personal constituency" which includes staff members and close personal contacts (Fenno 1977).

When thinking about home style as it relates to images, specifically images that members of Congress post on social media, we argue that members of Congress have the greatest incentives to use images in this context as a means of appealing to their *re–election constituency* and provide evidence for this using two illustrative examples of analyses from data produced by convolutional neural networks. In the first analysis, we identify objects in each of the 296,460 Facebook photos of members of Congress that we collected using a pre–trained convolutional neural network classifier and explore how object usage differs among Democratic and Republican members of Congress. We find that objects which best distinguish Democratic from Republican members of Congress are related to the military. In the second analysis, we train a convolutional neural network classifier to identify the race of individuals in images posted by members of Congress and explore how Democratic and Republican MCs represent African–Americans in the photos that they post. We find that Democrats tend to significantly over–represent African–Americans in the photos that they post, while only Southern Republicans tend to sig-

nificantly under–represent African–Americans. Findings from both of these analyses suggest that members of Congress use images as a means of communicating with their re–election constituencies.

5.1 Example 1: Objects in the images of Democratic and Republican members of Congress

Our first analysis demonstrates how a pre–trained convolutional neural network model, used to identify objects in images, can be used to study how members of Congress present themselves through images. The model that we used to analyze the images in our database is a *Resnet50* architecture model with *ImageNet* weights. The *Resnet50* architecture is a 152 layer convolutional neural network model developed at *Microsoft Research* (He et al. 2016). The architecture was the winner of the 2015 ImageNet Large Scale Visual Recognition Competition (ILSVRC) and achieved an average error rate of 3.57% on a test set containing thousands of ImageNet objects. The error rate of an image classifier is simply the % of images incorrectly classified and is the standard means through which image classifiers are evaluated and compared (Krizhevsky, Sutskever, and Hinton 2012).

The pre–trained *ResNet50* model that we used was implemented in the Python package *Keras* was chosen because of its superior ability to identify ImageNet objects[6]. Of the objects that the classifier can identify, politically

[6]A database of the object classes can be found online here: http://image-net.org/challenges/LSVRC/2014/browse-synsets

Pre–Processing Step	**Description**
Image loading.	*Load full color image and resize to* $224x224$.
Image to tensor.	*Convert image into 3D tensor of pixel intensities (RGB) from each channel.*
Tensor formatting	*Further tensor formatting/prep for image classification.*

Table 5 – Pre-Processing steps for identifying objects using the *ResNet50* convolutional neural network architecture in *Keras* with pre-trained *ImageNet* weights

relevant objects include flags, labeled as "flagpole" or "flagstaff," objects related to the military ("military uniform", "warplane") and handguns ("revolver", "six–gun", "six–shooter"), among others. Using this pre–trained classifier, we identified objects in each of the photos for each of the members of Congress in our database.

Image data, like text data, also requires pre–processing before it is presented to the classfier as outlined in Table 5. The first step involves resizing images so that they are all the same size. In this case images were resized to $224x224$ because this is the size that the *Resnet50* model is optimized for. The second step involved conversion of the loaded and resized image to a three–dimensional tensor array of pixel intensity values and the third step involves concatenating these tensors with other relevant image information such as class labels in preparation for classification.

Once the image is presented to the *ResNet50* classifier, the classifier estimates a probability distribution over all of the object categories. From this probability distribution, the five highest probability objects contained within a photo were generated as in Figure 14. For analysis purposes, we only re-

tained classified objects for which the top estimated probability was above 50%.

Object	Probability
Military Uniform	0.77
Steel Drum	0.04
Assault Rifle	0.03
Drum	0.02
Banjo	0.01

(a) (b)

Figure 14 – **(a)**– Photo from the Facebook profile of Lindsey Graham (R–SC) the text post below the photo reads "It's been one of the great honors of my life to serve in the Air Force in some capacity for more than three decades. This summer, I'll turn 60 the mandatory retirement age for the Air Force Reserves..."; **(b)** Top five predicted objects by probability from the *Resnet50* convolutional neural network classifier.

After classifying objects in each of the photos, high–probability classified objects for each member of Congress was stored as text data. In order to understand which objects best distinguish Democratic from Republican members of Congress, a random forest model was estimated using the document–term matrix of the objects for each member of congress and "Republican" or "Democrat" as a class label.

Figure 15 contains a plot of objects which contributed most to the random forest classifier's ability to identify Republican members of Congress. The measure of object importance here is the % increase in the mean squared error (MSE) of the model when that variable (object) is shuffled and the model

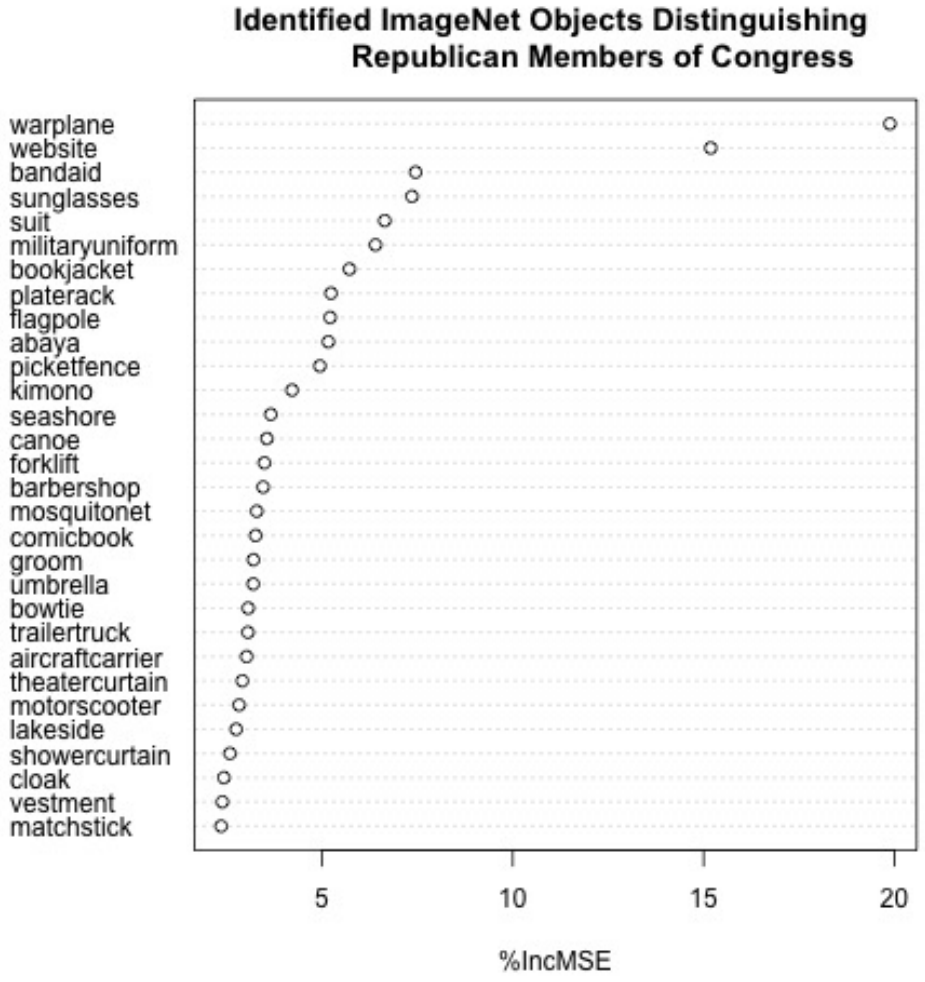

Figure 15 – Objects distinguishing Republican from Democratic members of Congress from an estimated random forest classifier model. Object importance here is measured as the % increase in the model's mean squared error (MSE) as the result of randomly shuffling that variable. Here we see that military objects ("warplane", "military uniform"),photos overlaid with text such as memes ("website"), male specific objects ("suit") and objects related to nationalism/patriotism ("flagpole"), tend to distinguish Republican from Democratic members of Congress.

is re–estimated. Here it is clear that military objects ("warplane", "military uniform"), photos overlaid with text such as memes ("website"), male specific objects ("suit") and objects related to nationalism/patriotism ("flagpole"), do the best job of distinguishing Republican from Democratic members of Congress. Since veterans and current members of the armed services tend to overwhelmingly identify as Republican (Teigen 2007), usage of military objects among Republicans provides evidence suggesting that Republicans use images to communicate with their re–election constituencies.

5.2 Example 2: Racial Representation in House Members Images

In this illustrative example, we train a convolutional neural network classifier using the VGG16 architecture described above to identify the race of people in images that members of Congress pose with in their Facebook photos and explore racial representation in a sub–sample of the photos that we collected which contain faces. We accomplished training in three steps. First, we estimated weights for a VGG16 architecture model using 61,500 training images which contain the race annotations: white, black, hispanic and Asian provided with the datasets as ground truth. The training images consist of a combination of two databases with labeled facial images. The first database is a collection of 17,500 labeled PubFig (Kumar et al. 2009) images and the second database is a collection of 44,000 labeled images from American

high school yearbooks. The Pubfig image database is an excellent initial source of labeled faces because the faces in the database contain a wide variety of angles, lighting and poses, allowing for robust training. The labeled yearbook database was collected and annotated by co–authors Crystal Lee and Shiry Ginosar and provides a convenient and diverse source of facial images. Weights were estimated using 100,000 epochs of training.

Figure 16 – Faces in yearbook and Facebook images detected using Viola-Jones algorithm with Haarcascades, cropped and converted to grayscale as a pre-processing step.

Images containing faces were then identified in the entire collection of Congressional Facebook images described above using the facial detection algorithm Haarcascades (Gorbenko and Popov 2012; Lienhart and Maydt 2002). A random 5% sample of photos containing faces from each member of Congress were then taken and these photos were further partitioned into photos containing only faces, again using the haarcascades algorithm as seen

in Figure 16. This resulted in a total of 33,777 cropped facial photos. Since training the classifier on the PubFig and Yearbook datasets and testing on Facebook portraits constitutes a domain shift that hurt the classification performance, we improved the classification accuracy via a process of bootstrapping by manually verifying the high–confidence race classifications and adding these into the training set. After these additions, we further fine-tuned our classifier by training 20,000 epochs on the augmented training set. The final average error rates were 10% for whites, 15% for blacks, 25% for Asians and 35% for hispanics.

5.2.1 Racial representation in Congress members' photos

After classifying the race of all individuals in House and Senate members' Facebook profiles, we explore comparisons between MCs Facebook profile "demographics" and compare these demographics to Congressional district demographics from American Community Survey (ACS) 5-year estimates. As mentioned above, we expect that members of Congress will use the photos that they post to on Facebook to appeal to their re–election constituencies. Thus, what we expect to find when we *compare Facebook profile demographics to district or state demographics* is that minority groups, such as African–Americans will be well represented or over–represented in the Facebook photos of Democratic members of Congress, since African–Americans are important part of Democrat's re–election constituency, while African–Americans should be under–represented in the Facebook photos of Republican members

of Congress since they are unlikely to constitute a significant enough part of their re–election constituency. In the figures below, "representation" for representative i is simply the difference between the % black estimated from MCs Facebook photos ($\%B_F$) and the % black in their Congressional district ($\%B_D$):

$$R_i = \%B_{Fi} - \%B_{Di} \tag{15}$$

We begin our analysis by exploring the distribution of R_i among white Democrats and Republicans in the House of Representatives. From Figure 17 it appears as if Democrats tend to over–represent African–Americans in their Facebook photos ($R_i > 0$) while Republicans do not. To explore this further, we estimated two models. The first model is a logistic regression of a dichotomous transformed version of R_i such that if $R_i > 0$, $R_i = 1$ or 0 otherwise:

$$logit(E[R > 0|Democrat, X]) = \alpha + \beta Democrat + X\gamma + \epsilon \tag{16}$$

Parameter estimates and predicted probabilities from Equation 16 give us a sense of the conditional probability of over–representation of African–Americans among Democratic and Republican House members.

Predicted conditional probability estimates of over–representation produced from Model (3) in Table 6 are plotted in Figure 19. The average pre-

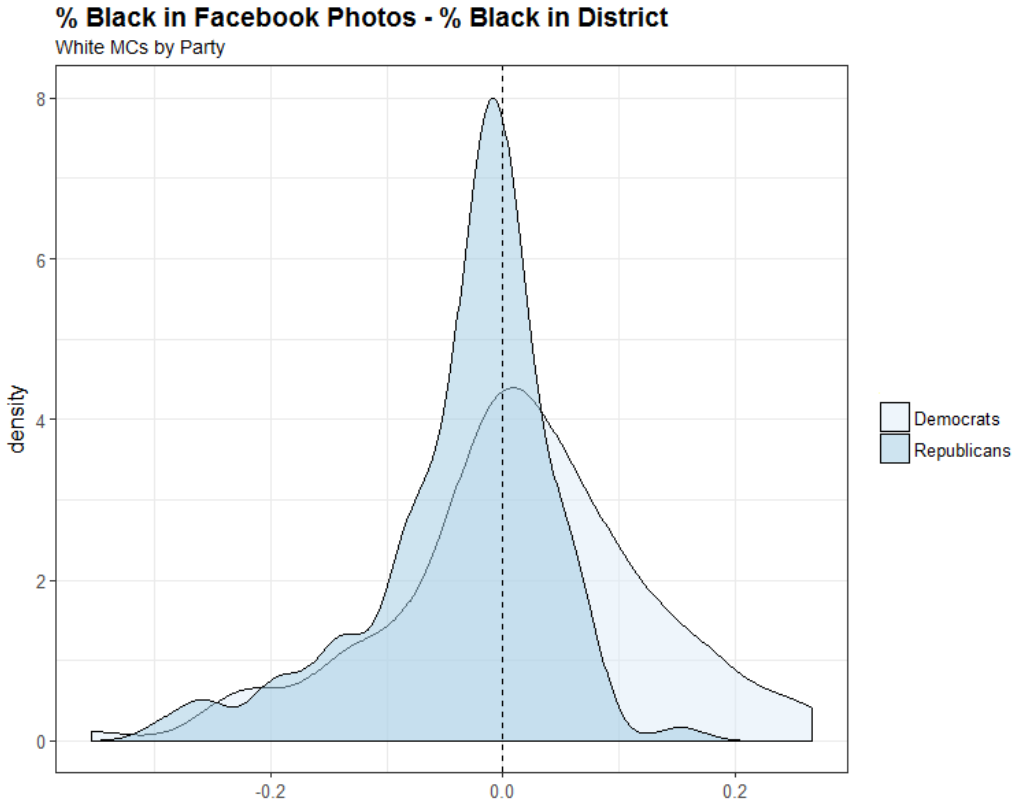

Figure 17 – Distribution of black representation R_i in Facebook photos among white Democratic and Republican House Members (% Black in Facebook Photos - % Black in district)

.

dicted conditional probability that Democrats will over–represent African–Americans in their Facebook photos[7] is 57.1% (52.1%, 62.2%) and for Republicans it is 34.7% (30.8%, 38.7%). Clearly, Democratic MCs, even after conditioning on race, region and other district demographics, are far more likely to over–represent African–Americans in their photos than are Republi-

[7]95% confidence intervals in parentheses

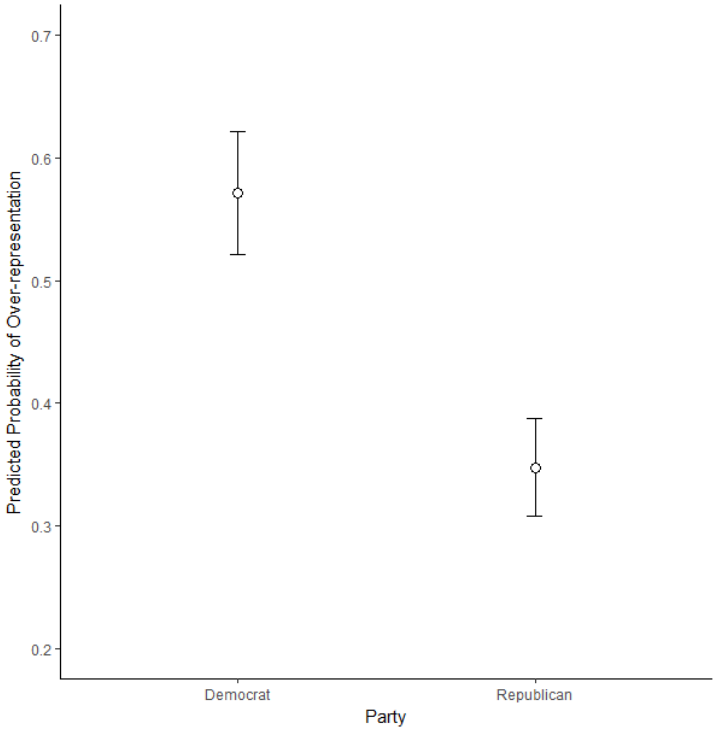

Figure 18 – Predicted probability of black over–representation in Facebook photos among Democratic and Republican House members from the logistic regression model: $logit(E[R > 0|Democrat, X]) = \alpha + \beta Democrat + X\gamma + \epsilon$

cans. Indeed, a closer look at the predicted conditional probability *distributions* among Democrats and Republicans (Figure 19) reveals more interesting information.

While predicted probabilities of over–representation among Democrats is unimodal with a center clearly higher than Republicans, Republicans' predicted probabilities are bimodal with one group of Republicans centering around 50% and other group of Republicans centering around 11%. This observation led us to an exploration of African–American *under–representation*

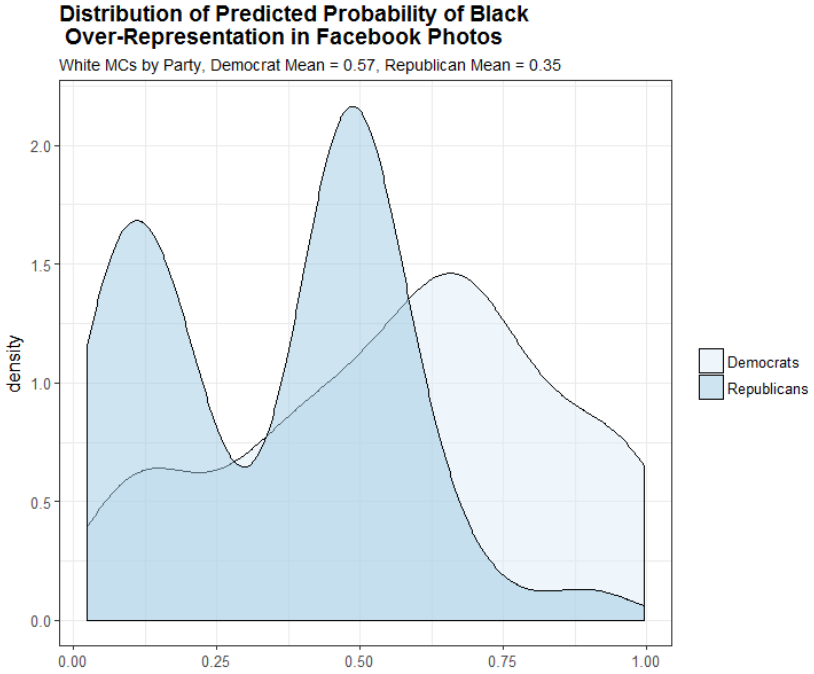

Figure 19 – Distributions of the predicted probability of black over–representation in Facebook photos from the logistic regression model: $logit(E[R > 0|Democrat, X]) = \alpha + \beta Democrat + X\gamma + \epsilon$

in Facebook photos among Democrats and Republicans. In addition to over–representation of African–Americans among Democrats, Republicans, especially those in districts which are very conservative or those districts in which constituents may harbor racial resentment (Feldman and Huddy 2005) may strategically *under–represent* African Americans. Given the South's history of racial tensions (Acharya, Blackwell, and Sen 2016), we might expect Southern House members, especially those in more conservative districts to *under–represent* African–Americans in their photos at a higher rate than House members outside of the South. Figure 20 contains estimates of the %

DV: 1 = %Black in Photos > %Black in District, 0 = Otherwise			
	(1)	(2)	(3)
Democrat	0.918***	0.803**	0.849*
	(0.272)	(0.305)	(0.406)
White MC	.	-0.311	-2.056**
		(0.380)	(0.696)
South	.	.	-1.641***
			(0.410)
% White in District	.	.	6.462***
			(1.659)
% Hispanic in District	.	.	5.796***
			(1.380)
N	230	230	230
R^2	0.058	0.081	0.371

$*p \leq 0.05 ** p \leq 0.01 *** p \leq 0.001$

Table 6 – Logistic regression of "Over-representation" on MCs party and other covariates. Coefficents reported are raw coefficient values estimated from the equation: $logit(E[R > 0|Democrat, X]) = \alpha + \beta Democrat + X\gamma + \epsilon$. Odds ratio calculations ($exp(Democrat)$) suggest that Democratic MCs are more than twice as likely ($exp(0.849)$) to over-represent African–Americans in their photos.

of MCs that under–represent African–American's in their photos by party and region. While both Democrats and Republicans in the South tend to under–represent African–Americans, it is clear that Southern Republicans have much higher rates of black under–representation of all groups at 88.7% of Southern Republicans under–representing African–Americans in their photos.

To confirm this result conditional on relevant district and member covariates, we estimated the following logistic regression model in which the dependent variable was coded as 1 if the MC under–represented African–

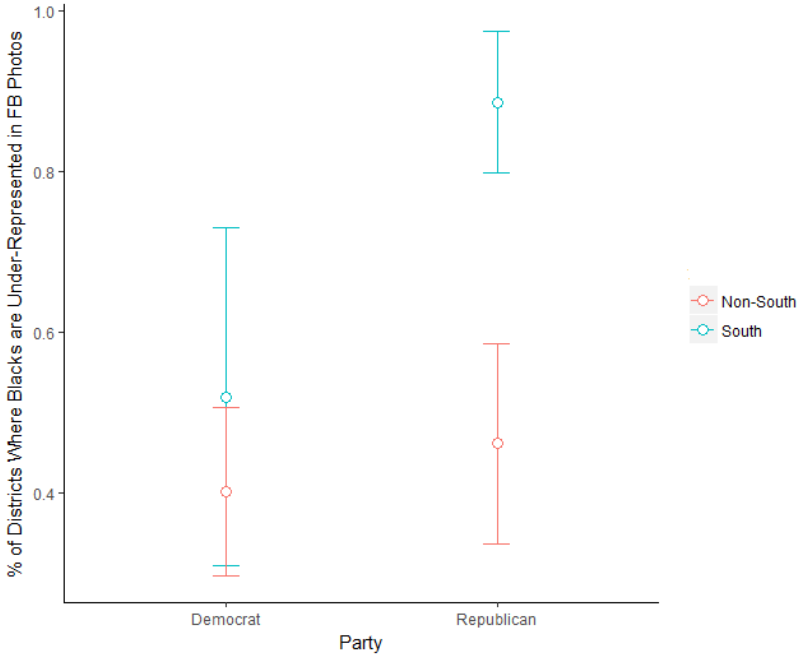

Figure 20 – % of MCs in which African–Americans are under–represented in Facebook photos in Southern and Non-Southern districts. There are large, statistically significant differences in under–representation among Republicans in Southern and Non-Southern districts with 88.7% of Southern Republicans under–representing African–Americans.

Americans in their photos, *Republican* is party affiliation of the MC and *South* is whether the MC's district is in a Southern state:

$$\begin{aligned} logit(E[R < 0|Republican, South, X]) =& \alpha + \beta_1 Republican + \beta_2 Republican\ x\ South \\ &+ \beta_3 South + X\gamma + \epsilon \end{aligned} \tag{17}$$

Table 7 model (3) contains raw coefficient estimates from Equation 17.

DV: Black under–representation in photos.

	(1)	(2)	(3)
Republican	0.918***	0.242	0.410
	(0.272)	(0.331)	(0.407)
Republican x South	.	1.737*	1.885*
		(0.677)	(0.833)
South	.	0.476	-0.070
		(0.456)	(0.665)
District % Black	.	.	10.211***
			(2.317)
White MC	.	.	2.744**
			(0.882)
District % Hispanic	.	.	-2.927*
			(1.303)
N	230	230	230

$*p \leq 0.05 ** p \leq 0.01 *** p \leq 0.001$

Table 7 – Raw coefficient estimates of the logistic regression of "under–representation" on MCs Republican party affiliation and Southern district. Coefficents reported are raw coefficient values estimated from the equation: $logit(E[R < 0|Republican, South, X]) = \alpha + \beta_1 Republican + \beta_2 Republican\ x\ South + \beta_3 South + X\gamma + \epsilon$. Odds ratio calculations ($exp(Republican)$) suggest that Republican MCs in the South are over 6 times more likely ($exp(1.885) = 6.6$) to under-represent African–Americans in their photos.

Conditional on member and district covariates, we find that Southern Republicans are about 6.6 times more likely [8] than all other House members to under–represent African–Americans in their photos. There are several explanations for this finding. One possibility is that Southern Republicans simply don't pose with African–Americans by chance. This explanation seems rather unlikely for several reasons. First, African–Americans comprise a sig-

[8] $exp(\beta_2) = exp(1.885) = 6.6$

nificantly higher % of most districts in Southern states than non-Southern states, yet under–representation is far less common among Republicans in non–Southern states (46.2% of districts v. 88.7% of districts in Southern states). Second, we saw from results in Table 7 after conditioning on % black in the MCs district, the interaction term β_2 on *Republican x South* actually *increased.* Another, more likely explanation for this phenomenon is that Republican MCs in Southern states, which have historically have high rates of racial resentment among whites, may seek to minimize any possibility of antagonizing constituents who harbor racial resentment or feelings of racial animosity. While a more detailed discussion of this claim is beyond the scope of this paper, we hope that students of Congress explore this claim in more depth in the future.

6 Discussion

The use of images by political figures to manipulate public opinion and sentiment is by no means a new phenomenon. Shrewd political figures such as Lyndon Johnson recognized the potential that images had to shape how they were perceived by the public and accordingly appointed the first White House photographer, Yoichi Okamoto, to do just that during his term in office. At the same time, however, photographs taken by brave journalists and chroniclers of the Vietnam War laid bare the horrors and devastation of a war which eventually led to Johnson's steep decline in popularity and his

eventual decision to not run for a second term.

While the use of images to achieve political ends is not new, our ability to systematically study and understand how, when and why they are used has only recently been made possible by the prolific use of images by politicians, political organizations and the public via the internet and recent developments in computer vision which allow for fast and accurate identification of complex features from labeled image data. Here, we take advantage of both developments in an effort to provide a broad framework for political image analysis using these techniques and simultaneously demonstrate how they can be used to understand the modern relevance of home style (Fenno 1978) with two illustrative examples. By breaking down images into their simplest political elements: objects and people, our framework provides a basis from which scholars can explore which aspects of images are used by politicians and political organizations the purpose of communicating with the public.

In addition to this, we demonstrate how neural network techniques can be used to understand home style through images. Using convolutional neural networks and empirical analysis of photos posted by members of the House and Senate on their Facebook profiles, we provide evidence that MCs use images to communicate with their re–election constituencies. Neural networks hold a tremendous amount of promise as a means of systematically understanding how images are used as a means of political communication in the digital age. Despite their promise, however, the black box nature of these methods remain a challenge to unlocking their full potential for so-

cial research purposes. Fortunately, recent developments in neural network research are beginning to unpack the black box with techniques such as "attention weights" which allow for a better understanding of how these very complex classification methods work (Xu et al. 2015).

References

Accetti, Carlo Invernizzi, and Fabio Wolkenstein. 2017. "The crisis of party democracy, cognitive mobilization, and the case for making parties more deliberative." *American Political Science Review* 111 (1): 97–109.

Acharya, Avidit, Matthew Blackwell, and Maya Sen. 2016. "The political legacy of American slavery." *The Journal of Politics* 78 (3): 621–641.

Baker, Andy. 2015. "Race, paternalism, and foreign aid: Evidence from US public opinion." *American Political Science Review* 109 (1): 93–109.

Barberá, Pablo. 2015. "Birds of the Same Feather Tweet Together. Bayesian Ideal Point Estimation Using Twitter Data." *Political Analysis* 23 (1): 76–91.

Bond, Robert, and Solomon Messing. 2015. "Quantifying social medias political space: Estimating ideology from publicly revealed preferences on Facebook." *American Political Science Review* 109 (01): 62–78.

Dilliplane, Susanna, Seth K Goldman, and Diana C Mutz. 2013. "Televised exposure to politics: New measures for a fragmented media environment." *American Journal of Political Science* 57 (1): 236–248.

Esterling, Kevin M, David MJ Lazer, and Michael A Neblo. 2013. "Connecting to constituents: the diffusion of representation practices among Congressional websites." *Political Research Quarterly* 66 (1): 102–114.

Feldman, Stanley, and Leonie Huddy. 2005. "Racial resentment and white opposition to race-conscious programs: Principles or prejudice?" *American Journal of Political Science* 49 (1): 168–183.

Fenno. 1978. *Home style: House members in their districts*. HarperCollins.

Fenno, Richard F. 1977. "US House members in their constituencies: An exploration." *American Political Science Review* 71 (3): 883–917.

Giger, Nathalie, and Heike Klüver. 2016. "Voting against your constituents? How lobbying affects representation." *American Journal of Political Science* 60 (1): 190–205.

Gorbenko, Anna, and Vladimir Popov. 2012. "On face detection from compressed video streams." *Applied Mathematical Sciences* 6 (96): 4763–4766.

Hainmueller, Jens, and Dominik Hangartner. 2013. "Who gets a Swiss passport? A natural experiment in immigrant discrimination." *American Political Science Review* 107 (01): 159–187.

Hayes, Danny, Jennifer L Lawless, and Gail Baitinger. 2014. "Who cares what they wear? Media, gender, and the influence of candidate appearance." *Social Science Quarterly* 95 (5): 1194–1212.

He, Kaiming, 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*. pp. 770–778.

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.

Hebb, Donald Olding. 1949. *The organization of behavior: A neuropsychological approach*. John Wiley & Sons.

Herrnson, Paul S. 1992. "Campaign professionalism and fundraising in congressional elections." *The Journal of Politics* 54 (3): 859–870.

Hirsch, Alexander V. 2016. "Experimentation and persuasion in political organizations." *American Political Science Review* 110 (1): 68–84.

Hopkins, Daniel J. 2015. "The upside of accents: Language, inter-group difference, and attitudes toward immigration." *British Journal of Political Science* 45 (03): 531–557.

Horiuchi, Yusaku, Tadashi Komatsu, and Fumio Nakaya. 2012. "Should candidates smile to win elections? An application of automated face recognition technology." *Political Psychology* 33 (6): 925–933.

Hutchings, Vincent L, Hanes Walton Jr, and Andrea Benjamin. 2010. "The impact of explicit racial cues on gender differences in support for confederate symbols and partisanship." *The Journal of Politics* 72 (4): 1175–1188.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "Imagenet classification with deep convolutional neural networks." In *Advances in neural information process-*

ing systems. pp. 1097–1105. https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

Kumar, Neeraj, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. 2009. "Attribute and Simile Classifiers for Face Verification." In *In IEEE International Conference on Computer Vision (ICCV)*. http://acberg.com/papers/kbbn09iccv.pdf.

Lawson, Chappell, Gabriel S Lenz, Andy Baker, and Michael Myers. 2010. "Looking like a winner: Candidate appearance and electoral success in new democracies." *World Politics* 62 (04): 561–593.

LeCun, Yann A, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. "Efficient backprop." In *Neural networks: Tricks of the trade*. Springer.

Lienhart, Rainer, and Jochen Maydt. 2002. "An extended set of haar-like features for rapid object detection." In *Image Processing. 2002. Proceedings. 2002 International Conference on*. Vol. 1 IEEE pp. I–900. https://pdfs.semanticscholar.org/72e0/8cf12730135c5ccd7234036e04536218b6c1.pdf.

Lopes da Fonseca, Mariana. 2017. "Identifying the Source of Incumbency Advantage through a Constitutional Reform." *American Journal of Political Science*.

Mattes, Kyle, and Caitlin Milazzo. 2014. "Pretty faces, marginal races: Predicting election outcomes using trait assessments of British parliamentary candidates." *Electoral Studies* 34: 177–189.

Mayhew, David R. 1974. *Congress: The electoral connection.* Yale University Press.

McCulloch, Warren S, and Walter Pitts. 1943. "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics* 5 (4): 115–133.

Prior, Markus. 2013. "Visual political knowledge: A different road to competence?" *The Journal of Politics* 76 (1): 41–57.

Rosenblatt, Frank. 1958. "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological review* 65 (6): 386.

Ryan, Timothy J. 2017a. "How Do Indifferent Voters Decide? The Political Importance of Implicit Attitudes." *American Journal of Political Science.*

Ryan, Timothy J. 2017b. "No compromise: Political consequences of moralized attitudes." *American Journal of Political Science* 61 (2): 409–423.

Sen, Maya, and Omar Wasow. 2016. "Race as a Bundle of Sticks: Designs that Estimate Effects of Seemingly Immutable Characteristics." *Annual Review of Political Science* 19: 499–522.

Teigen, Jeremy M. 2007. "Veterans' party identification, candidate affect, and vote choice in the 2004 US presidential election." *Armed Forces & Society* 33 (3): 414–437.

Tingley, Dustin. 2014. "Face-Off: Facial Features and Strategic Choice." *Political Psychology* 35 (1): 35–55.

Tomandl, Dirk, and Andreas Schober. 2001. "A modified general regression neural network (MGRNN) with new, efficient training algorithms as a robust black box-tool for data analysis." *Neural networks* 14 (8): 1023–1034.

Tomasello, Michael, Tricia Striano, and Philippe Rochat. 1999. "Do young children use objects as symbols?" *British Journal of Developmental Psychology* 17 (4): 563–584.

Vapnik, Vladimir Naumovich, and Vlamimir Vapnik. 1998. *Statistical learning theory*. Vol. 1 Wiley New York.

Wamsley, Gary L, and Mayer N Zald. 1973. "The political economy of public organizations." *Public Administration Review*: 62–73.

Wilson, James Q. 1974. *Political organizations*. Princeton University Press.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. "Show, attend and tell: Neural image caption generation with visual attention."

In *International Conference on Machine Learning*. pp. 2048–2057. https://arxiv.org/abs/1502.03044.