Visualizing neuron role with dimensionality reduction of feature space

Justine Zhang
CS 231N Project

Abstract

Convolutional neural networks are hard to understand. While conventional machine learning algorithms usually rely on a set of readily interpretable features, it is often unclear how such semantic meanings map to a set of neurons, and to the different layers of a network.

In this project, we attempt to better understand the action of neurons and layers by capturing the most important neurons and most important features that are learned at each layer. While a naive approach that examines single neurons yields readily interpretable features, we suggest that neurons may not be the fundamental semantic unit in a layer and that relative "importance" of a neuron is hard to quantify. We next suggest a latent variable approach, using singular value decomposition to view neurons and the images they classify in a low-rank space. We show that interpretable features arise when examining sets of images corresponding to high singular values, and that interpretability is loosely correlated with the magnitude of the singular value. We evaluate this new approach to visualization. Finally, using the latent variable strategy, we make some observations about the layer-by-layer behaviour of a pretrained CaffeNet model.

1. Introduction

Conventional machine learning algorithms generally rely on the process of feature extraction. Input data $\mathcal{X}$ is mapped to some feature space Φ consisting of reasonably interpretable features, and the feature mapping ϕ is then used to perform various tasks such as making predictions. In a classification problem, for instance, we may train a linear model that outputs some prediction $h(w \cdot \phi(x))$ on example x.

Convolutional neural networks have achieved strong results in the problem of image classification, but suffer from the drawback of uninterpretability. While high level claims can be made about the role of each neuron and each layer, it is often unclear how a particular neuron contributes to the classification task, and how the structure produces the rich set of features learned. Having knowledge of the relationship between network architecture and features could allow one to better tune the structure of a network to produce better classification accuracies, and allow one to better understand the salient features of a dataset.

More concretely, we would like to answer two questions:

- What image features result from a specific neuron or group of neurons?
- Within each layer, which neurons produce the most salient features?

1.1. Notation

We establish some notation that will be used throughout the report. Suppose we have a neural network with layers $L_1, \dots L_k$ with number of neurons $n_1, \dots n_k$. Each layer L_i induces a feature mapping $\phi_i : X \to \mathbb{R}^{n_i}$ where $\phi_i(x)_j$ is the activation of the jth neuron in L_i corresponding to input data x, for $j = 1, \dots n_i$. We also establish that X consists of n images $x_1, \dots x_n$.

1.2. Overview of approach

Our general approach is to produce and study the images which trigger high activations on specific neurons. We therefore infer the features associated with the neurons to be the features which are common to these images. More specifically, we consider the activations of neurons in a pretrained network on a dataset of images; for this project we use CaffeNet and the 2012 Imagenet validation set.

This preliminary analysis suggests that while images with high activations in select neurons have interpretable features in common, many distinct neurons produce redundant sets of images and ranking the features yielded in terms of relative importance is not straightfoward. Additionally, performing analyses based on previous literature suggests that neurons may not be the most fundamental unit of semantics, since high-scoring images in linear combinations of neurons also yield interpretable results. This motivates us to suspect that the feature space is lower rank and that we should consider multiple neurons in aggregate.

Our more refined approach is therefore to perform truncated singular value decomposition on the feature space,

and consider high-scoring images in latent dimensions corresponding to high singular values. Our results show that while the magnitude of the singular value roughly corresponds to feature importance, more work needs to be done to more thoroughly capture the features revealed by the network.

2. Related Work

Most of the previous work in understanding neural networks has relied on capturing behaviours of individual neurons. On the neuron level, the simplest method is to examine images which produce high activations on target neurons; images could be taken from datasets or generated via gradient descent to maximize neuron activation, resulting in interpretable features (see Erhan et. al, 2009). In Lee et. al (2008), kth layer neurons were visualized by taking a linear combination of the filters in the $k-1$st layer which are most strongly connected to them; the method found that the second layer captured information about contours and corners in an image. Zeiler and Fergus (2013) present an approach which uses a deconvolutional network attached to the original network to project activations of neurons at later layers back to the input pixel space. This method revealed some general problems with the original architecture of the AlexNet model studied, such as clear aliasing due to large strides; the visualizations hence informed changes to the AlexNet architecture which produced performance gains.

Szegedy et. al (2014) suggest, however, that there is no distinction between individual high-level neurons and combinations of neurons in producing interpretable features. The paper demonstrated random projections of ϕ at various layers which produced high-scoring images that produced features that were as interpretable as those produced while examining single neurons, suggesting that semantic information is contained in the space of neurons, rather than in single neurons. Much of our analysis is motivated by this idea.

3. Approach

At a high level, we attempt to characterize features learned by a trained network $N = L_1, \dots L_k$ by examining subsets of the corresponding validation dataset $\mathcal{X}$. We proceed to outline specific methods used:

3.1. Single-neuron "naive" method

First, we examine images which produced particularly high, or particularly low activations on particular neurons. Specifically, for neuron j in layer i, we find and manually examine images in

$$\text{argmax}_{x \in \mathcal{X}} |\langle \phi_i(x), e_j \rangle|$$

where e_j is the jth standard basis vector (in general, we will overload argmax to mean images for which the above inner product is *close* to maximum). We select neurons to examine in two ways. First, we consider neurons with high variance in activations across the image dataset:

$$\text{argmax}_j \text{Var}_{x \in \mathcal{X}}(\langle \phi_i(x), e_j \rangle)$$

Intuitively, since these neurons produce more variable activations for each image, we expect them to yield more discriminating features. Hence, variance could loosely correspond to "importance" of a neuron in representing a feature which plays a larger role in the layer. As a baseline, we also perform the same procedure on randomly-selected neurons.

3.2. Random projection method

Next, we consider the method presented in Szegedy et. al. At layer i, we find and manually examine images in

$$\text{argmax}_{x \in \mathcal{X}} |\langle \phi_i(x), v \rangle|$$

where v is a random vector in $\mathbb{R}^n$. As in the first method, we consider variance as a proxy for importance, in this case of the random direction in which ϕ_i is projected. Hence, we set an empirically-determined cutoff of $\text{Var}_{x \in \mathcal{X}} \langle \phi_i(x), v \rangle \geq 850$, which takes the tail of the distribution of such variances across a sample of random vectors v. We also randomly select directions without constraint as a baseline.

3.3. Singular value decomposition method

Next, we consider a low-rank representation of the space of features. Specifically, consider matrix $\Phi_i \in \mathbb{R}^{n_i, n}$ with columns $\phi_i(x)$ for $x \in \mathcal{X}$. Hence, $\Phi_{i(j,m)} = \langle \phi_i(x_m), e_j \rangle$ for $j = 1, \dots n_i$ and $m = 1, \dots n$. Intuitively, we can consider neurons as objects with activation on each image as a feature.

Let $\tilde{\Phi}_i$ be Φ with columns normalized to mean 0 and variance 1. We consider the low-rank approximation produced by truncated singular value decomposition:

$$\tilde{\Phi}_i \approx \Phi_{i,s} = U_{i,s} S_{i,s} V_{i,s}^T$$

where s is the lower rank, $S_{i,s} \in \mathbb{R}^{s,s}$ is the diagonal matrix consisting of the top s singular values of Φ_i, $U_{i,s} \in \mathbb{R}^{n_i, s}$ has columns consisting of left singular vectors of Φ_i corresponding to the top s singular values, and $V_{i,s} \in \mathbb{R}^{s,n}$ has columns consisting of the right singular vectors. We will denote each component as U, S, V for convenience.

Given this decomposition, we can now examine images which have especially high or low values in each latent dimension. In particular, for dimension $j = 1, \dots s$ we ,manually examine images in

$$\text{argmax}_{m=1,\ldots n}|\langle V_j, e_m\rangle|$$

where V_j is the jth right singular vector.

This method can be seen as an extension of the previous random projection method, in that truncated singular value decomposition considers directions which produce the highest variance of features - in this case, activations on neurons. Our initial hypothesis is that high singular values yield images with distinctive or significant features within each layer, whereas lower singular values may correspond to less significant or interpretable features.

For each layer, we use scikit-learn's randomized svd function to perform truncated SVD. We choose the number of singular values to retain by enforcing an empirical threshold of approximately 200, such that all singular values $\sigma < 200$ were dropped.

4. Experiments

In this section, we detail some results.

4.1. Architecture and inputs

We perform our experiments on the CaffeNet architecture, using the ILSVRC2012 classification task validation set as our image dataset. CaffeNet consists of 5 convolutional layers followed by 3 fully-connected layers. Our analysis is repeated on all 8 layers; specifically, for each convolutional layer, we study the activations produced by the last layer in the conv-relu-pool-(norm) pipeline.

Out of the 50000 images in the validation set, we only consider activations produced by the 28000 images which are correctly classified by CaffeNet. Of these correct images, we choose a random subset of 6000 to allow our analysis to fit in memory.

4.2. Focused analysis of pool5 layer

Due to relatively interesting initial results, we start by performing a more focused analysis of the activations in the fifth convolutional layer after pooling, referred to below as "pool5". The layer contains $256\times6\times6 = 9216$ neurons (we call numpy's reshape function to flatten the 3-D structure of the layer). Having noticed little difference in distribution of variance or images produced either way, we will use the normalized activations (as in $\tilde{\Phi}$) for the entire analysis, for consistency.

4.2.1 Single neuron analysis

We first perform the naive method detailed above. Initially, we examine the distribution of variances of the (normalized) neuron activations. A histogram of activation variances can be found in Figure 1.

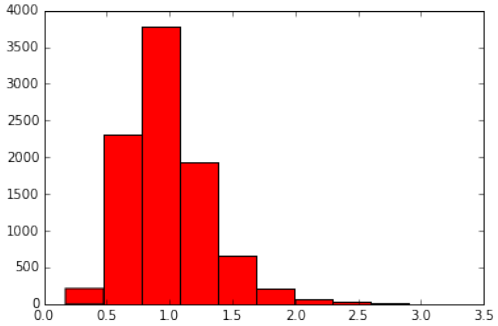

Figure 1. Histogram of variance of activations produced by images in dataset for each neuron in pool5.

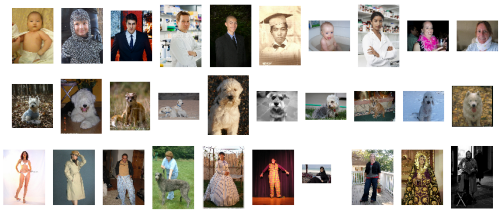

Figure 2. Images with the highest activations on neurons with high activation variance, for pool5 layer. From up to down, inferred features are: humans with focus on face, white fluffy dogs, standing humans.

With our initial view that neuron importance and feature interpretability roughly correspond to activation variance, the distribution of variances may suggest that a small subset of the neurons in pool5 with top variance may contain the most salient features in the layer. Indeed, examining the sets of images with highest activations on the top 5 neurons by variance, we see that many of these sets are unified by readily interpretable features, as seen in Figure 2.

However, further analysis suggests that there may be very little correspondence between feature interpretability and neuron activation variance. For instance, sets of images with high activations on randomly selected neurons often have clear features in common as well, even though the activation variance of that neuron is much lower, as shown in Figure 3. We note that for variances less than around 0.5, the features produced become much less interpretable, but for variances above this threshold, the relationship is coarse at best.

A second problem emerges: different neurons often produce similar-looking features. Within the high-variance neurons, several correspond to sets of images which are almost identical to the sets displayed in Figure 1. Additionally, the sets of images which produced exceptionally low activations on many neurons were near-identical. This suggests that the space of features as learned by single neurons is highly redundant. This casts further doubt on the

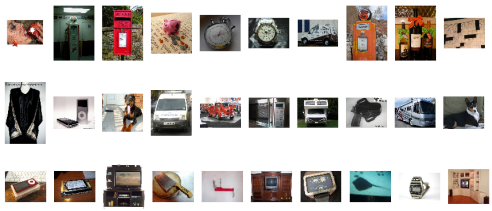

Figure 3. Images with the highest activations on random neurons, for pool5 layer. From up to down, inferred features are: objects with text, white and black rectangular objects, compact rectangular objects.

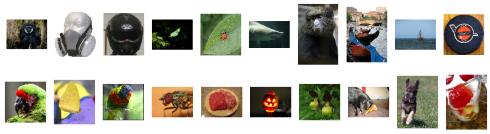

Figure 4. High-scoring images for selected random projections. Top to bottom: helmet and other rounded head shapes, parrot colours.

idea that individual neuron activation variance corresponds to feature saliency, since many neurons with smaller activation variances still yielded similar features to neurons with larger activation variances.

It would make sense that variance in feature value has some correlation to feature saliency - indeed, methods such as PCA look for high-variance directions in feature space. However, in the case of convolutional neural networks, we suspect that this relation is complicated by two factors. First, especially for earlier layers, localization as enforced by the stride architecture means that adjacent neurons often behave in similar ways. Second, because networks are deep, a low-variance neuron that is closely connected to a high-variance region in a higher layer may still have a more discriminative role in the classification process than a neuron in a high-variance region within a single layer.

4.2.2 Random projection method

We next turn to the random projection method. Figure 4 shows some of the more interpretable sets of images which score highly on projections in a series of random directions.

While some random projections do yield somewhat interpretable sets of images, the relationship between these images is more spurious than for single neurons. Additionally, as with the single neuron case, we note very little difference in interpretability from high-variance projections to lower-variance projections, and indeed, it wasn't clear whether random projections, or projections filtered for high variance, produced more interpretable sets of images.

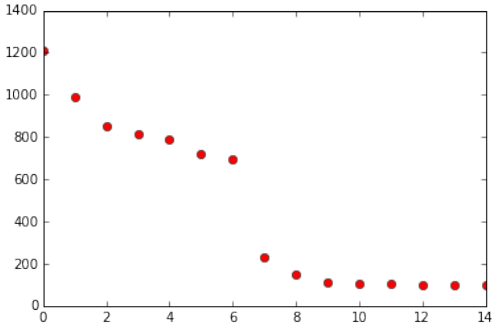

Figure 5. Singular values for features in pool5.

4.2.3 Singular value decomposition method

We now apply the truncated SVD method to the features in pool5. Figure 5 displays a graph of the first fifteen singular values of Φ_5; in light of this distribution, we set a target rank of 8.

As detailed above, for each of the top singular values, we consider the set of images with the highest and lowest values in the corresponding right singular vector. These images are displayed in Figure 6.

Interestingly, this method produces highly interpretable features for each of the dimensions corresponding to the highest singular values. Additionally, among the high singular values, none of the features which emerge are redundant, as was the case for the naive method on single neurons. This suggests that truncated SVD reduces the redundancy of the feature space quite effectively, as desired. While it is hard to qualify relative importance or interpretability of a feature, especially among the first few dimensions, we also note that dimensions corresponding to lower features yield images whose relationship is unclear. Notably, for dimensions beyond the 8th (not shown), the sets of top-scoring images are quite similar, suggesting that further dimensions are less well-separated. Conversely, this is some assurance that gauging the quality of a feature based on its "interpretability" by a single human being, while sketchy, isn't completely baseless, if we see some correspondence between human interpretability and a quantity such as the magnitude of the singular value.

Another interesting observation is that while high-scoring images for particular neurons are often actually of similar objects, whereas the sets of images produced via the SVD method tend to span multiple completely separate classes and instead share some abstract characteristic. For instance the top images in the fourth dimension are all yellow objects, and the bottom images in the seventh dimension are all obelisk-shaped objects, be it actual obelisks or long bird necks. This suggests that perhaps the SVD method captures more "fundamental" features learned at

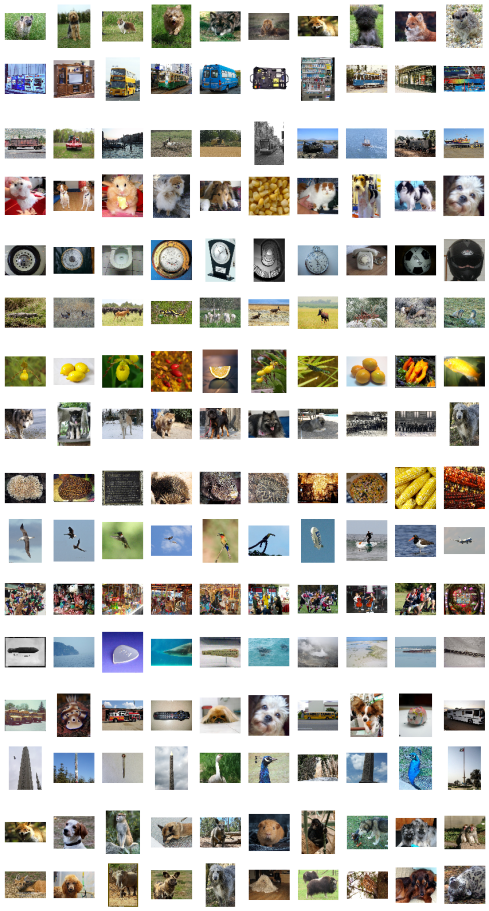

Figure 6. Top images for the top 8 latent dimensions in pool5. Images at the top correspond to dimensions with larger singular values than images at the bottom. Each successive pair of rows corresponds to a latent dimension, with the first row consisting of images with high values in that dimension and the second row consisting of images with low values.

that layer (although how to quantify and validate this statement is unclear).

Some notable pitfalls of the SVD method can also be seen. First, while the magnitude of singular values drops off significantly after the top few, the low-rank representation captures the feature space relatively poorly in terms of thoroughness. Intuitively, given the rich diversity of features that arise from our first two methods, it seems dubious that each layer can be reduced to the 7 or 8 features (technically, x2 because we count high and low values) that arise out of the SVD method. To quantify this, we consider the reconstruction error of the low-rank approximation, given as the Frobenius norm of the difference between the approximation and the original matrix Φ:

$$\|U_s S_s V_s^T - \Phi\|_2$$

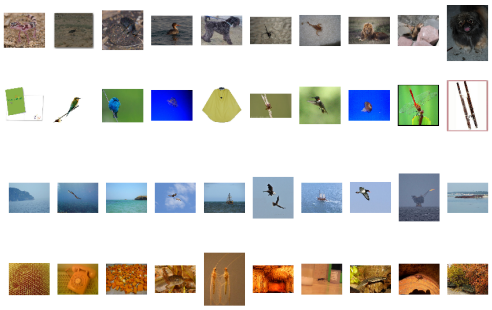

Figure 7. Top and bottom images for largest singular values for conv-relu-pool-norm1.

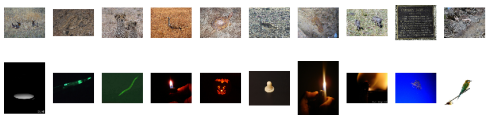

Figure 8. Top and bottom images for largest singular values for conv-relu-pool-norm2.

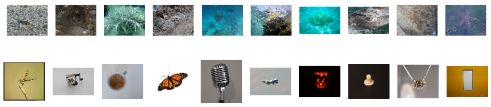

Figure 9. Top and bottom images for largest singular values for conv-relu3.

For $s = 8$, this value is 7050. (Normalizing for the number of entries in Φ produces a slightly less frightening result.)

Finally, as the redundancy of the top images in the later dimensions suggests, the features yielded by the SVD method are limited by the number of large singular values for Φ. We could theoretically envision characterizing the features accounted for in a layer as linear combinations of the SVD features, although it is an unclear task to assemble together separate visual features in this way.

4.3. Applying the SVD method to each layer

We conclude by applying the SVD method to each of the 8 layers of CaffeNet. For each layer, high and low scoring images in dimensions corresponding to singular values above a cut-off of approximately 200 are displayed in Figures 7-13 (pool5 already shown). For fully connected layers, since there are more large singular values, only a subset of image sets are shown.

We make some general remarks about each layer. First, we note that earlier layers tend to yield more abstract features such as colour and texture than later ones; some of the image sets for the fully-connected layers correspond quite precisely to image categories (e.g. snails). Next, we

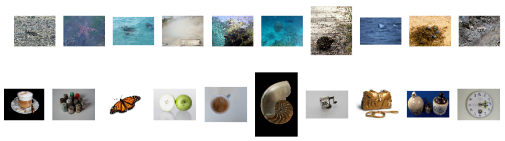

Figure 10. Top and bottom images for largest singular values for conv-relu4.

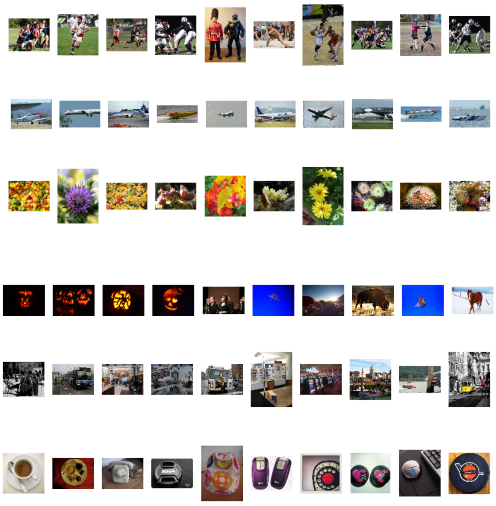

Figure 11. Top and bottom images for selected singular values for fc6.

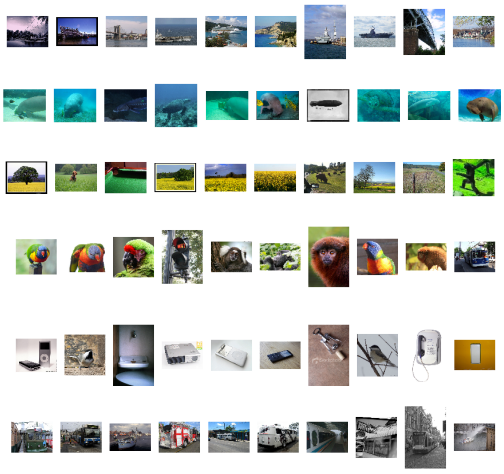

Figure 12. Top and bottom images for selected singular values for fc7.

note that earlier layers have less large singular values than later layers - indeed, the early conv layers only have one large singular value, compared to the fully connected layers, which have at least 8 each. While further study is needed to explain this result, we suggest that this is because earlier conv layers tend to learn more local features corresponding

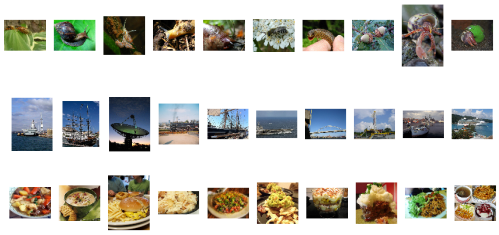

Figure 13. Top and bottom images for selected singular values for fc8.

to regions of the image (as determined by the convolution stride), whereas fully-connected layers tend to yield more global features.

5. Conclusion

We study various ways of visualizing salient features produced at each layer of a convolutional neural network, via examining subsets of images from a corresponding dataset. Using CaffeNet and the ILSRVC2012 classification task validation set as examples, we consider three methods of visualization: a naive neuron-by-neuron method, a random projection method, and a low-rank method via truncated singular value decomposition. Our results on the fifth convolutional layer suggest that the richness of features represented by each layer exceeds that which can be captured by each method, although certain quantities such as variance and magnitude of singular value can be treated as rough proxies for determining feature importance. Finally, using the SVD method, we visualize features for each of the CaffeNet layers, and suggest that features represented at each layer become progressively less localized and less abstract.

5.1. Future work

A lot more work can be done in understanding how convolutional neural networks behave via visualizing represented features (especially given that we admittedly dragged our feet in starting this project). This project suggests a few particular possibilities. First, the fact that globally important features are difficult to classify suggests that layers might actually represent a rich set of localized features, which could be characterized. One possibility is to use a clustering algorithm to discern groups of redundant or similar neurons, and infer features from each cluster; the suspicion that a layer's feature space consists of several localized features suggests that agglomerative clustering might be fruitful. Exploring the degree to which features capture local as opposed to global characteristics of an input image could also be fruitful, and a study of features could attempt to correlate similar features with the actual coordinates of neurons within each layer. Finally,

examining how features interact across layers would also be useful. As mentioned above, we suspect that importance of neurons as informed by activation variance might have been obscured by the fact that a high-variance neuron might forward-propagate into a low-variance region in a later layer. Applying the ideas in Lee et. al could result in more insights.

References

[1] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *Dept. IRO, Université de Montréal, Tech. Rep*, 2009.

[2] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area v2. In *Advances in neural information processing systems*, pages 873–880, 2008.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[5] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.