International Workshop on Machine Learning in Visualisation for Big Data (2020)
D. Archambault, I. Nabney and J. Peltonen (Editors)

# Visual Analysis of the Impact of Neural Network Hyper-Parameters

Daniel Jönsson[1], Gabriel Eilertsen[1], Hezi Shi[2], Jianmin Zheng[2], Anders Ynnerman[1], and Jonas Unger[1]

[1]Linköping University, Institute for Science and Technology, Sweden
[2]Nanyang Technological University, Institute for Media Innovation, Singapore

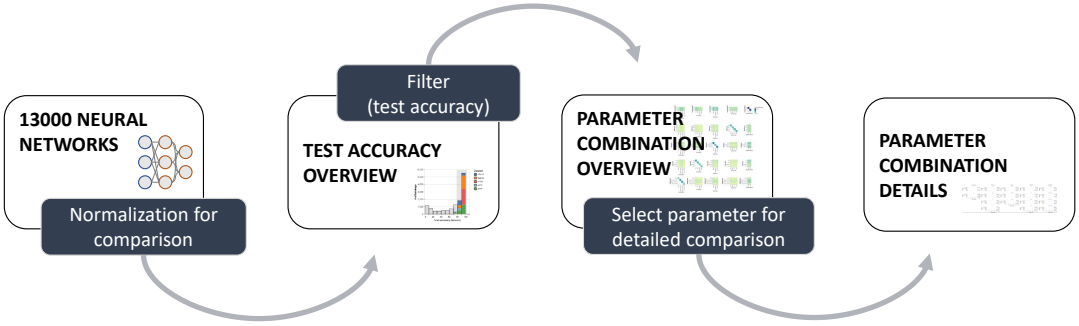



**Figure 1:** *Overview of the visual analysis pipeline used to understand and discover the effect hyper-parameters have on neural network model performance. The test accuracy overview provides means for selecting good or bad network trainings. The most significant parameter combinations involved in the filtered selection can be detected through a heat-map matrix. Further analysis of the impact of a selected parameter with respect to all other parameters is enabled through small multiples plots of parameter setting aggregates.*

**Abstract**

*We present an analysis of the impact of hyper-parameters for an ensemble of neural networks using tailored visualization techniques to understand the complicated relationship between hyper-parameters and model performance. The high-dimensional error surface spanned by the wide range of hyper-parameters used to specify and optimize neural networks is difficult to characterize – it is non-convex and discontinuous, and there could be complex local dependencies between hyper-parameters. To explore these dependencies, we make use of a large number of sampled relations between hyper-parameters and end performance, retrieved from thousands of individually trained convolutional neural network classifiers. We use a structured selection of visualization techniques to analyze the impact of different combinations of hyper-parameters. The results reveal how complicated dependencies between hyper-parameters influence the end performance, demonstrating how the complete picture painted by considering a large number of trainings simultaneously can aid in understanding the impact of hyper-parameter combinations.*

**CCS Concepts**

• ***Computing methodologies*** → ***Neural networks;*** • ***Human-centered computing*** → ***Visual analytics;***

## 1. Introduction

The last decade has seen a surge in usage of neural networks for solving a wide variety of tasks, from medical diagnosis to language translation. For example, deep convolutional neural networks (CNNs) enable powerful modeling of natural image tasks such as classification, by learning complex relations through a layered structure of learnable weights. However, the optimization of neural networks is challenging, where millions of weights should be updated by observing data. Thus, the tuning of the training setup and the optimization procedure is of critical importance, and is predominantly a manual effort. The tuning is performed by selecting a number of hyper-parameters, for example to decide on model behavior, initial values of weights, and optimization strategy [PBB19]. While recent development in deep learning has brought forward techniques that can successfully optimize deep neural networks in a wide range of situations, there is still significant room for performance improvement by fine-tuning the hyper-



DOI: 10.2312/mlvis.20201101

parameters. However, it is not feasible to exhaustively explore all possible combinations of hyper-parameters for each case. Instead, we must understand their interconnected behavior to come up with best-practices.

The problematic nature of hyper-parameter exploration can be explained from the highly non-convex error surface spanned by the hyper-parameters, where the different hyper-parameters can be in tight dependence on each other. Also, many hyper-parameters are categorical and not suitable for gradient-based optimization. Thus, it is not feasible to use conventional methods for optimizing and exploring hyper-parameters, which is further emphasized by the fact that random searches have proven to be more efficient [BB12]. While hyper-parameter search aims at finding the optimal combination of parameters [BBBK11], our goal is to provide a better understanding of the impact of different hyper-parameters in terms of model performance. For this purpose, we use a visual analysis approach to explore the differences over multiple dimensions of hyper-parameters, for thousands of trained neural networks.

The main objective of this paper is to distill general knowledge about the relation between hyper-parameters and model performance through analysis of a large number of trained CNNs. Thus, in this work, we are less concerned about the details of a specific network. We do this by visually presenting statistics of the training performance of the *neural weight space* dataset [EJR*20]. Overview is provided through interactive distribution views, linked to a heat-matrix for parameter combination impact overview. Further details are presented in small-multiples plots of parameter setting aggregates for analysis and comparison of interconnected hyper-parameters.

The main questions investigated and answered by our analysis are: 1) how can we visually present statistics of the results from a large number of trained neural networks, and convey information that is not visible from incremental sampling of hyper-parameters as is done in conventional hyper-parameter optimization methods? and 2) which parameter combinations have the most positive or negative impact on the test accuracy, and how do different combinations relate to each other?

Throughout the paper, we use a broad definition of hyper-parameters, which include both architectural (size/depth, activation function), and optimization parameters (initialization, optimizer, batch size, etc.), and even comprising the dataset used for training.

## 2. Related Work

There is a large amount of existing work considering the importance of different hyper-parameters and approaches to optimize them [HHLB14,CdM15,vRH18], e.g. using techniques such as random search [BB12] and Bayesian optimization [BBBK11,Moc12]. While these methods mostly focus on optimizing the hyper-parameter selection and speeding up the training process, in this work we are more interested in understanding how different hyper-parameters relate to each other, and how visual analysis can aid in explaining such information.

Visual analysis has proven important in understanding and improving performance in deep learning [HKPC18]. For example, visualization is an important concept for explainable AI, which attempts to shed light on how neural networks operate [SVZ13,ZF14, YCN*15, SCD*17]. However, so far visualization as a tool to understand hyper-parameters has seen limited research, partly due to the large number of trainings required to get a detailed picture of the hyper-parameter space. Some previous works developed tools to support hyper-parameter search and analysis via interactive visual analytics [TSM*11,PBCR11,LCW*18] or for verifying hyper-parameter optimization performance [YRK*15]. A few works also look at comparative visualization of a larger number of models trained with different hyper-parameters [Bre15,HDK*19,EJR*20], but are limited in the number of models or hyper-parameter combinations compared.

In this work, we use visual support to reveal relations between hyper-parameter combinations applied to a large number of CNNs. We make use of the neural weight space dataset, see [EJR*20] for details, which provides 13K CNNs trained to perform classification with a diverse selection of hyper-parameters on 5 different image datasets. The hyper-parameters are randomly sampled, and include both architectural specifications and optimization parameters. Each network has been specified with 3-5 convolutional layers, three max-pooling layers, and 3-5 fully connected (FC) layers. Although there are many hyper-parameters in the dataset, we choose to look at a subset deemed to reveal the most interesting and complex relations. The subset includes:

| | |
|---|---|
| **Dataset** | MNIST [LBB*98], CIFAR-10 [KH09], SVHN [NWC*11], STL-10 [CNL11], Fashion-MNIST [XRV17] |
| **Batch size** | 32, 64, 128, 256 |
| **Augmentation** | Off, On |
| **Optimizer** | ADAM [KB14], RMSProp [HSS12], Momentum SGD |
| **Activation** | ReLU [NH10], ELU [CUH15], Sigmoid, TanH |
| **Initialization** | Constant, Random normal, Glorot uniform, Glorot normal [GB10] |

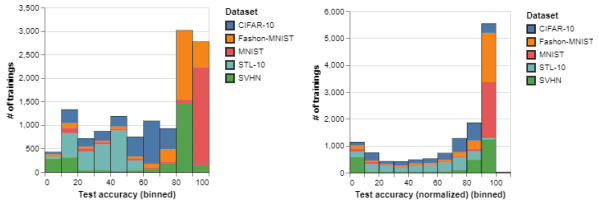
## 3. Visual Analysis Design

To facilitate exploration of the many parameters at the same time, we utilize overview and filter techniques combined with data aggregation and small multiples plots [Tuf01]. The visualization frameworks Vega-Lite [SMWH16], Plotly [Inc15], and Inviwo [JSS*19] have been used during the design process and for the implementation of the visual analysis tool illustrated in Figure 1. In the following, we will describe the flow of going from overview to details along with chosen visualization techniques.

### 3.1. Model Performance Overview

The two central parts in our exploration are input data, which largely affect the test accuracy, and the test accuracy itself. Thus, we provide an overview of these two central parts using a stacked histogram depicting the distribution of the test accuracy for the different data sets. As can be seen in Figure 2a, the test accuracy corresponding to each data set has its own mode (due to varying

complexity), which means that a straight forward distribution visualization does not allow for easy comparison across data sets. Therefore, to enable comparison across data sets, a linear normalization, $x' = \frac{x - x_{min}}{x_{max} - x_{min}}$, is applied on a per-data set basis, resulting in the histograms depicted in Figure 2b. The normalized test accuracy can thus be used for selection/filtering interaction of the test accuracy range to investigate good/bad parameter combinations.




**(a)** *Test accuracy.* **(b)** *Normalized test accuracy.*

**Figure 2:** *Depiction of test accuracy for different data sets using (a) no normalization and (b) per-data set linear normalization. The normalized distributions loosen the distribution skewing due to varying input data difficulties and provides easier parameter comparison.*

### 3.2. Parameter Combination Overview

Test accuracy range selection serves as a basis for exploring combinations of parameters. Here, a parameter combination overview, based on the current filter, is provided by a heat-map matrix. The perceptually linear multi-hue sequential colormap colors [Bre0x] in the heat-map matrix represent the aggregate number of networks included in the selection for each combination of two parameters. Thus, as an example and illustrated to the right in Figure 3; given a selection of high test accuracy the heat-map matrix elements depicting high counts indicate good parameter combinations. The aggregate number of networks can be used for comparison in this way since the parameter settings were initially randomly selected.

### 3.3. Parameter Combination Details

Once an interesting parameter combination has been found it may be of interest to see more details and understand if there are additional parameter settings affecting the test accuracy. Inspired by the Becker's Trellis barley plot [BCS96], a third view depicts the averages of the test accuracy for this purpose, see Figure 4. Here, each parameter is split by its options and a primary parameter, e.g., data set. This shows how the test accuracy varies depending on the combination of each parameter and the selected primary parameter. Additionally, a secondary parameter can be chosen and encoded as shape/color allowing for three parameters to be analyzed at the same time.

Note that other multidimensional visualization techniques such as parallel coordinates were evaluated but discarded since most of the parameters are not quantitative, which causes all lines to intersect at the same point. While this can partially be remedied by random offsets, as demonstrated in the supplementary material, it is still difficult to see complex interdependencies between multiple hyper-parameters.

## 4. Results

With the presented tool for visual analysis, it is possible to discover many different properties of how hyper-parameter combinations impact performance. Here, for space-limitation reasons, detailed information is provided for a few such findings. The most obvious pattern is the importance of good initialization, optimizer, and activation function, highlighting the impact development in optimization techniques has had during the last decade. For example, a modern initialization scheme (Glorot uniform/normal) and activation function (ReLU/ELU) has a profound effect on the success of training. However, in order to demonstrate more complex relations between hyper-parameters, which can be revealed with the aid of visual analysis, we will give some examples of less obvious nature. While these relations could potentially be found by training and testing individual combinations of hyper-parameters, this could be difficult without prior knowledge on what to search for, and the visual analysis of many trainings aid in discovering novel patterns.

**Augmentation** At a first inspection it seems that augmentation has a small impact on the results, which is also supported by the simpler correlation analysis in [EJR*20]. However, this does not reveal the complete picture, as the augmentation has different impacts depending on the other hyper-parameters and on the dataset. First of all, augmentation has less effect on the more "artificial" datasets (MNIST, Fashion-MNIST, SVHN). By selecting only the more difficult datasets of natural images, CIFAR-10 and STL-10, we can explore the distribution of hyper-parameter combinations across different test accuracies. If we only select the trainings centered around the main mode of performance, i.e. the most frequent training outcomes, "no augmentation" is most common, see Figure 3 (top). However, selecting only the top models, there is a clear benefit in using augmentation (Figure 3 (bottom)). This indicates that only the more difficult optimization problems effectively make use of augmentation and does so in a limited number of trainings. It is clear how a good combination of optimizer (ADAM), initialization (Glorot), and activation function (ELU) is essential to reach the top performance. At the same time, it is also not a guarantee that the top percentile of possible performance will be reached if these hyper-parameters are used.

**Optimizer** One of the most evident correlations between hyper-parameter and performance is the optimizer, where modern techniques such as ADAM and RMSprop clearly outperform a conventional momentum SGD optimizer. However, the situation is more complex than a global correlation; the performance of momentum SGD is tightly connected to the combination of other hyper-parameters, such as activation function and initialization scheme. Figure 4 shows two examples of how three different hyper-parameters relate to each other. In the left plot, it can be seen that while both ADAM and RMSProp generate results invariant to the batch size, the conventional momentum SGD creates a negative correlation between batch size and accuracy, i.e., the performance is better for smaller batch sizes. In the right plot, comparing optimizers, initialization schemes, and activation functions, the sensitivity of conventional momentum SGD to both initialization and activation function is revealed. For example, if ELU activation is used together with Glorot initialization, then this optimizer performs on

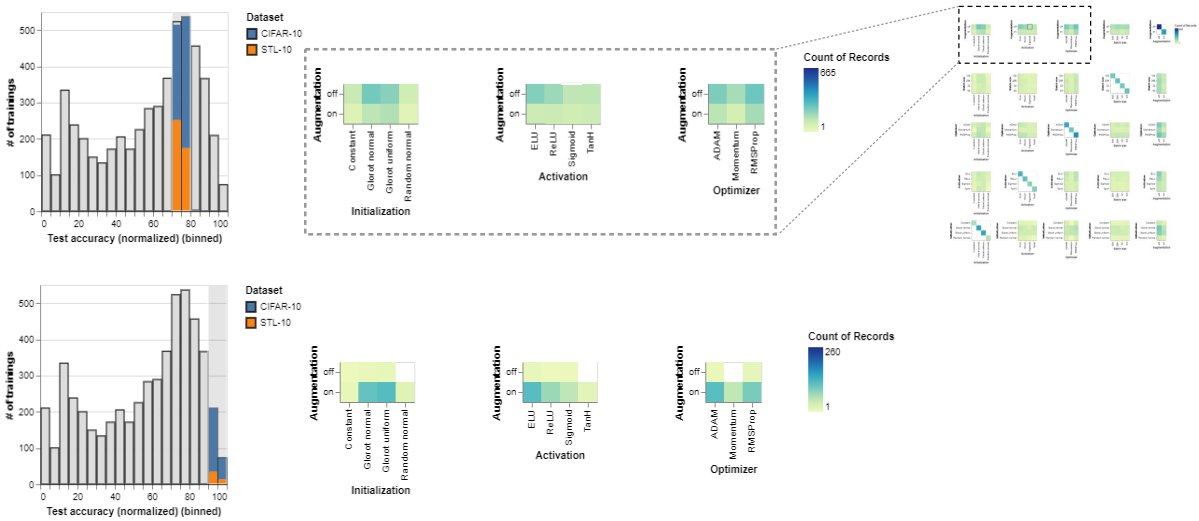



**Figure 3:** *Analysis of the distribution of hyper-parameters for different ranges of end test accuracy of CIFAR-10 and STL-10. (Top) the main mode of the accuracy distribution, indicated by the colored selection in the histogram, does not benefit from the augmentation, while the best performing models show the opposite pattern (bottom). The center images have been extracted from their respective heat-map matrix for presentation purposes.*

par with the state-of-the-art optimizers, whereas there is a large discrepancy between the optimizers when constant or random normal initialization is used. It should be noted that the momentum SGD optimizer has been used with a constant momentum term, and that the results could be different if this is tuned differently.

**Activation function** Initialization and activation function is a sensitive combination. While a Sigmoid function is inferior together with a naive intitialization – most trainings are even non-convergent if constant initialization is used – it is equally efficient as ReLU when Glorot initialization is used. In fact, Figure 4 (right) shows that Sigmoid performs slightly better than ReLU for the Glorot normal initialization scheme. Another interesting observation is how the ELU activation definition shows a consistent small improvement over the widely used ReLU in most circumstances.

## 5. Conclusions

We have introduced a set of tailored visual analysis tools for revealing detailed information on the interplay between neural network hyper-parameters. We used the tools to study the performance of 13K CNNs, and reported a selection of interesting dependencies between hyper-parameters. We saw how the most optimal combination of optimizer, activation function and initialization, in conjunction with augmented training data, is required to reach the highest accuracy on natural images. However, it is still only a fraction of the trainings with these settings that actually reach top performance.

While visual analysis is a powerful tool for exploring hyper-parameters, one of the main limitations is that it also runs the risk of revealing patterns that are not statistically significant. Thus, a possible direction for future work is to provide statistical analysis along with the visualizations [JBF*19]. Another direction is to analyze even more intricate dependencies between hyper-parameters

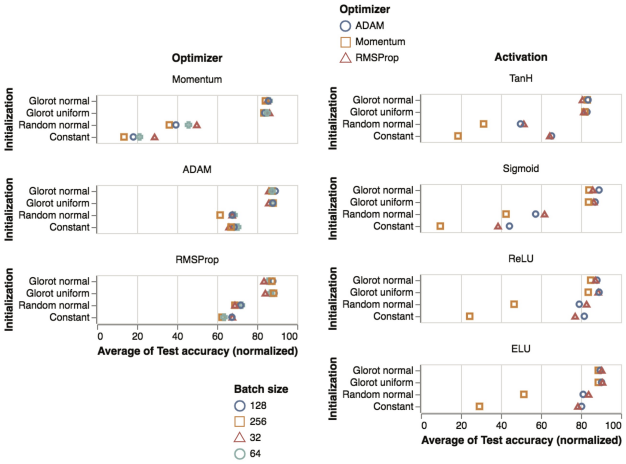



**Figure 4:** *Behavior of different optimizers for different initialization schemes, batch sizes, and activation functions. The plots reveal the sensitivity of conventional momentum SGD to batch size (left) and activation function (right), but the sensitivity is only apparent when a naive initialization scheme is used.*

using the tools presented. For this purpose, it would be of interest to explore a more automated visual analysis pipeline, where interesting patterns in the data can be found without manual effort. We believe that visual support for understanding the complicated multi-dimensional nature of hyper-parameters will be an important concept in machine learning, in both research and development applications.

**Acknowledgements** This project was supported by the Wallenberg Autonomous Systems and Software Program (WASP) and the strategic research environment ELLIIT.

## References


[BB12] BERGSTRA J., BENGIO Y.: Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13*, Feb (2012), 281–305. 2

[BBBK11] BERGSTRA J. S., BARDENET R., BENGIO Y., KÉGL B.: Algorithms for hyper-parameter optimization. In *Proceedings of the Advances in Neural Information Processing Systems* (2011), pp. 2546–2554. 2

[BCS96] BECKER R. A., CLEVELAND W. S., SHYU M.-J.: The visual design and control of trellis display. *Journal of Computational and Graphical Statistics 5*, 2 (1996), 123–155. 3

[Bre0x] BREWER C. A.: Colorbrewer 2.0, 200x. URL: http://www.ColorBrewer.org/. 3

[Bre15] BREUEL T. M.: The effects of hyperparameters on SGD training of neural networks. *arXiv preprint arXiv:1508.02788* (2015). 2

[CdM15] CLAESEN M., DE MOOR B.: Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127* (2015). 2

[CNL11] COATES A., NG A., LEE H.: An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (2011), pp. 215–223. 2

[CUH15] CLEVERT D.-A., UNTERTHINER T., HOCHREITER S.: Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* (2015). 2

[EJR*20] EILERTSEN G., JÖNSSON D., ROPINSKI T., UNGER J., YNNERMAN A.: Classifying the classifier: dissecting the weight space of neural networks. *arXiv preprint arXiv:2002.05688* (2020). 2, 3

[GB10] GLOROT X., BENGIO Y.: Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), pp. 249–256. 2

[HDK*19] HAMID S., DERSTROFF A., KLEMM S., NGO Q. Q., JIANG X., LINSEN L.: Visual ensemble analysis to study the influence of hyper-parameters on training deep neural networks. In *Machine Learning Methods in Visualisation for Big Data* (2019), The Eurographics Association. 2

[HHLB14] HUTTER F., HOOS H., LEYTON-BROWN K.: An efficient approach for assessing hyperparameter importance. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32* (2014). 2

[HKPC18] HOHMAN F. M., KAHNG M., PIENTA R., CHAU D. H.: Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics* (2018). 2

[HSS12] HINTON G., SRIVASTAVA N., SWERSKY K.: Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2

[Inc15] INC. P. T.: Collaborative data science, 2015. URL: https://plot.ly. 2

[JBF*19] JÖNSSON D., BERGSTRÖM A., FORSELL C., SIMON R., ENGSTRÖM M., YNNERMAN A., HOTZ I.: A Visual Environment for Hypothesis Formation and Reasoning in Studies with fMRI and Multivariate Clinical Data. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (Brno, Czech Republic, 2019), The Eurographics Association. doi:10.2312/vcbm.20191232. 4

[JSS*19] JÖNSSON D., STENETEG P., SUNDÉN E., ENGLUND R., KOTTRAVEL S., FALK M., YNNERMAN A., HOTZ I., ROPINSKI T.: Inviwo-a visualization system with usage abstraction levels. *IEEE Transactions on Visualization and Computer Graphics* (2019). 2

[KB14] KINGMA D. P., BA J.: ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 2

[KH09] KRIZHEVSKY A., HINTON G.: *Learning multiple layers of features from tiny images*. Tech. rep., Citeseer, 2009. 2

[LBB*98] LECUN Y., BOTTOU L., BENGIO Y., HAFFNER P., ET AL.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324. 2

[LCW*18] LI T., CONVERTINO G., WANG W., MOST H., ZAJONC T., TSAI Y.: Hypertuner: Visual analytics for hyperparameter tuning by professionals. In *Proceedings of the IEEEVIS 2018 Workshop on Machine Learning from User Interaction for Visualization and Analytics, Berlin, Germany* (2018). 2

[Moc12] MOCKUS J.: *Bayesian approach to global optimization: theory and applications*, vol. 37. Springer Science & Business Media, 2012. 2

[NH10] NAIR V., HINTON G. E.: Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning* (2010), pp. 807–814. 2

[NWC*11] NETZER Y., WANG T., COATES A., BISSACCO A., WU B., NG A. Y.: Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning* (2011). 2

[PBB19] PROBST P., BOULESTEIX A.-L., BISCHL B.: Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research 20*, 53 (2019), 1–32. 1

[PBCR11] PRETORIUS A. J., BRAY M., CARPENTER A. E., RUDDLE R. A.: Visualization of parameter space for image analysis. *IEEE Trans. Vis. Comput. Graph. 17*, 12 (2011), 2402–2411. 2

[SCD*17] SELVARAJU R. R., COGSWELL M., DAS A., VEDANTAM R., PARIKH D., BATRA D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR 2017)* (2017), pp. 618–626. 2

[SMWH16] SATYANARAYAN A., MORITZ D., WONGSUPHASAWAT K., HEER J.: Vega-Lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2016), 341–350. 2

[SVZ13] SIMONYAN K., VEDALDI A., ZISSERMAN A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013). 2

[TSM*11] TORSNEY-WEIR T., SAAD A., MÖLLER T., HEGE H., WEBER B., VERBAVATZ J.: Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. Vis. Comput. Graph. 17*, 12 (2011), 1892–1901. 2

[Tuf01] TUFTE E. R.: *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 2001. 2

[vRH18] VAN RIJN J., HUTTER F.: Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018). 2

[XRV17] XIAO H., RASUL K., VOLLGRAF R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017). 2

[YCN*15] YOSINSKI J., CLUNE J., NGUYEN A., FUCHS T., LIPSON H.: Understanding neural networks through deep visualization. In *In ICML Workshop on Deep Learning* (2015). 2

[YRK*15] YOUNG S. R., ROSE D. C., KARNOWSKI T. P., LIM S.-H., PATTON R. M.: Optimizing deep learning hyper-parameters through an evolutionary algorithm. In *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments* (2015), pp. 1–5. 2

[ZF14] ZEILER M. D., FERGUS R.: Visualizing and understanding convolutional networks. In *European Conference on Computer Vision* (2014), Springer, pp. 818–833. 2