Semantic Bottleneck Layers: Quantifying and Improving Inspectability of Deep Representations

Max Losch[1] Mario Fritz[2] Bernt Schiele[1]
[1]Max Planck Institute for Informatics, Saarland Informatics Campus
[2]CISPA Helmholtz Center for Information Security. Saarland Informatics Campus
Saarbrücken, Germany
[1]*firstname.lastname@mpi-inf.mpg.de*
[2]*firstname.lastname@cispa.saarland*

Abstract

Today's deep learning systems deliver high performance based on end-to-end training but are notoriously hard to inspect. We argue that there are at least two reasons making inspectability challenging: (i) representations are distributed across hundreds of channels and (ii) a unifying metric quantifying inspectability is lacking. In this paper, we address both issues by proposing supervised and unsupervised Semantic Bottleneck (SB) layers we integrate into pretrained networks to align channel outputs with individual visual concepts and introduce the model agnostic AUiC metric to measure the alignment. We present a case study on semantic segmentation to demonstrate that SBs improve the AUiC up to four-fold over regular network outputs, while recovering state of the art performances.

1. Introduction

While single output loss training (end-to-end) is key to top performance of deep learning – it is also the main obstacle to obtain inspectable systems. A key problem is that all intermediate representations are learned without interpretability as explicit objective leaving them opaque to humans. Furthermore, assessing inspectability has remained a fairly elusive concept since its framing has mostly been qualitative (e.g. saliency maps). Given the increasing interest in using deep learning in real world applications, interpretability and a quantification of such is critically missing.

Desiderata for inspectability. To address this, we demand information in each channel to be represented by a single semantic (sub-)concept, similarly to how the task of classification enforces semantic meaning on the output predictions. This is derived from a simple observation: distributed representations do not lend themselves to trivial interpretation. Hence, we desire to adapt deep networks to reduce distributed representations by (i) reducing the number of channels to a minimum, (ii) associating them with semantic (sub-)concepts, and, at the same time, (iii) aiming to lose as little overall performance as possible. In our view such semantics based inspectability can be seen as a way towards achieving true interpretability of deep networks.

Our contributions are three-fold. Firstly, we introduce two network layers we term Semantic Bottlenecks (SB) based on linear layers to improve alignment with semantic concepts by (i) supervision to visual concepts and (ii) regularizing the output to be one-hot encoded. Secondly, we show that integrating SBs into a state-of-the-art architecture does not impair performance, even for low-dimensional SBs that reduce the number of channels from 4096 to 30. Finally, we introduce the novel AUiC metric to quantify alignment between channel outputs and visual concepts for any model and show our SBs improve the baselines up to four-fold. Based on our knowledge, we are first to show such a modular approach that substantially improves inherent model inspectability without losing performance of state-of-the art classification models. Combined with our AUiC, our approach is general and easy to apply to any model.

2. Related Work

As argued in prior work [5], interpretability can be largely approached in two ways. The first being post-hoc interpretation, for which we take an already trained and well performing model and dissect its internals a-posteriori to identify important input features via attribution or grouping [1, 7, 8, 10, 11, 14]. The second approach is to design inherently interpretable models, either with [3, 4] or without supervision [6]. In contrast to our work, these models are generally not designed to be modular for application in modern classification architectures and are challenging to integrate or fail to reach good performance. In order to investigate the inspectability of deep networks, Bau et al. proposed NetDissect - a method counting number of channels associable to single visual concepts [1]. Our AUiC metric leverages the ideas of NetDissect and extends it to satisfy three criteria we deem important for measuring inspectability – which NetDissect does not satisfy.

Broden+ object	Sky	Building	Person	Road	Car	Lamp	Bike	Van	Truck	Motorbike	Train	Bus
# subordinate parts	1	5	14	1	9	3	4	6	2	3	5	6
Broden+ material	Brick, Fabric, Foliage, Glass, Metal, Plastic, Rubber, Skin, Stone, Tile, Wood											

Table 1: Relevant concepts from *Broden+* for the Cityscapes domain. Material concepts in bottom row and parts are grouped by their respective parent object (top 2 rows).

3. Semantic Bottlenecks

To approach more inspectable intermediate representations we demand information in each channel to be represented by a single semantic concept. We propose two variants to achieve this goal: (i) supervise single channels to represent a unique concept and (ii) enforce one-hot outputs to encourage concept-aligned channels and inhibit distributed representations. We construct both variants as layers that can be integrated into pretrained models, mapping intermediate representations to a semantic space. We name these supervised and unsupervised Semantic Bottlenecks (SB).

Case Study. To show the utility of SBs, we choose street scene segmentation on the Cityscapes dataset [2]. We use PSPNet [12] based on ResNet-101.

3.1. Supervised Semantic Bottlenecks (SSBs)

Variant (i) supervises each SB channel to represent a single semantic concept using additional concept annotations. One (or multiple) linear layers receive the distributed inputs from a host model and are supervised using an auxiliary loss to map them to target concepts. These predictions are concatenated and fed into the next layer of the host model.

Choosing concepts for Cityscapes. For our supervised SB-layer we choose concepts based on task relevancy for Cityscapes. *Broden+* [9] is a recent collection of pixel level annotations including parts and materials. We select 70 concepts we deem task relevant (see table 1).

Implementation details. Since the Broden concepts are not defined on the Cityscapes images, we train SSBs on a pretrained host which parameters are kept fix. We insert the SSB at two different locations: block4 and pyramid. We choose single 1×1-*conv* layers as classifiers for this task, which are integrated into the host network after training by adjusting the input dimensionality of the next layer. After integration, all downstream layers are finetuned.

3.2. Unsupervised Semantic Bottlenecks (USBs)

Clearly, the requirement for additional annotation and the uncertainty regarding choice of concepts is a limitation of SSBs. To address this limitation, we investigate the use of an annotation free method to (i) reduce number of channels, (ii) increase semantic association and (iii) lose as little performance as possible. To approach point (ii) we propose unsupervised semantic bottlenecks (USBs) that enforce *non*-distributed representations by approaching one-hot encodings. In the following we investigate the use of softmax activations as a means to address this point.

3.2.1 Construction of USBs

We keep using the same bottleneck framework as for SSBs, but add a softmax activation function on the outputs of 1×1-*conv* layers that we regularize accordingly to achieve one-hot encodings. Parameterizing softmax with a temperature T, it approaches $\arg\max$ when $T \to 0$. In our setup, we start with a high T, e.g. $T_0 = 1$ and reduce it to $T_\tau = 0.01$ in τ training iterations to approach $\arg\max$.

Implementation details We start with a pretrained PSPNet, integrate the USB and finetune all downstream layers plus the USB itself. During inference, we compute $\arg\max$ instead of softmax to acquire one-hot encodings.

4. Quantification of Layer Inspectability

We present the *AUiC* metric enabling architecture agnostic benchmarking, measuring alignments between channels and visual concepts. We specify three criteria that AUiC has to satisfy: (i) it must be a scalar measuring how well a set of visual concepts can be aligned to channel outputs. (ii) The metric must be model agnostic to allow comparisons between two different activation functions. (iii) The quantification must be computed unbiased w.r.t. the concept area in the image. The fundamental ideas inspiring our metric are based on the frequently cited NetDissect method [1].

4.1. AUiC metric

Our proposed metric involves two steps.

Channel-Concept matching. As first step, each channel needs to be identified as detector for a single concept. Given dataset $\mathbf{X}$ containing annotations for concept set C, we compare channel activations A_k and pixel annotations L_c, where $c \in C$. Since a channel output A_k is continuous, it needs to be binarized with a threshold θ_k acquiring binary mask $M_k \equiv M(k, \theta_k) = A_k > \theta_k$. Comparison can subsequently be quantified with a metric like $\text{IoU}(\mathbf{x}) = \frac{|M_k \cap L_c|}{|M_k \cup L_c|}$, $|\cdot|$ being cardinality of a set. A few things need to be considered w.r.t. to our criteria. IoU penalizes small areas more than large ones, since small annotations are disproportionally more susceptible to noise, which would become an issue later during optimizing θ. We address this issue using the *mean* IoU of positive *and negative* responses to balance the label area by its complements $\overline{M_k}$ and $\overline{L_c}$. The alignment score between channel and concept is subsequently defined over the whole dataset $\mathbf{X}$:

$$\text{mIoU}_{k,c}(\mathbf{X}) = \frac{1}{2}\left(\frac{\Sigma_{\mathbf{x}\in\mathbf{X}}|M_k \cap L_c|}{\Sigma_{\mathbf{x}\in\mathbf{X}}|M_k \cup L_c|} + \frac{\Sigma_{\mathbf{x}\in\mathbf{X}}|\overline{M}_k \cap \overline{L}_c|}{\Sigma_{\mathbf{x}\in\mathbf{X}}|\overline{M}_k \cup \overline{L}_c|}\right). \quad (1)$$

We sum over all samples before computing the fraction to include samples not containing concept c.

Secondly, the alignment between channel and concept is sensitive to θ_k. We keep the determination of θ_k agnostic to the activation distribution by finding critical point $\theta^*_{k,c}$ – now per channel and concept – maximizing $\text{mIoU}_{k,c}(\mathbf{X}, \theta_{k,c})$ – now parameterized with the threshold:

$$\theta^*_{k,c} = \arg\max_{\theta_{k,c}} \text{mIoU}_{k,c}(\mathbf{X}, \theta_{k,c}). \quad (2)$$

channel	trained concept	IoU assignment	Our mIoU assignment
16	person/hair	torso (0.07) painted (0.06)	person (0.57) hair (0.55)
32	lamp/shade	painted (0.07) brown (0.06)	shade (0.58) lamp (0.53)
18	person/foot	torso (0.07) black (0.05)	person (0.53) foot (0.53)
69	wood material	brown (0.07) painted (0.07)	wood (0.53) floor (0.52)

Table 2: Channel identification comparison for SSB@pyramid using either IoU or mIoU. The latter reducing size bias substantially.

This leaves $|C|$ concepts per channel, for which we identified the best thresholds. The final assignment is performed in a last step, choosing concept c^* maximizing mIoU

$$c^* = \arg\max_c \text{mIoU}_{k,c}(\mathbf{X}, \theta^*_{k,c}). \tag{3}$$

Each concept can be assigned to multiple channels, but not vise versa.

Scalar quantity. The second step involves summarizing the identifiability to a scalar value – 0 indicating no channel can be identified and 1 all. Given a global mIoU threshold ξ we can determine the fraction of channels having a greater mIoU. In order to keep the metric agnostic to the choice of ξ, we define the final AUiC as the AUC under the indicator function – counting identifiable channels – for all $\xi \in [0.5, 1]$:

$$\text{AUiC} = 2 \int_{0.5}^{1} \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}_{\text{mIoU}_{k,c^*} \geq \xi} \, d\xi. \tag{4}$$

4.2. Discussion

We conclude by showing that AUiC satisfies our three criteria and delineate it to the related NetDissect-measure.

Clear definition in $[0, 1]$. 0 must indicate no channel alignment – 1 perfect alignment for all channels. AUiC satisfies this criteria as it integrates over all mIoU thresholds. NetDissect instead chooses a specific IoU threshold $\xi = 0.04$ giving a false sense of security since all channels only require to pass this threshold.

Agnostic to model. To enable comparison across diverse types of models, we require a metric agnostic to the distribution of outputs. AUiC satisfies this criteria since it considers the threshold $\theta^*_{k,c}$ that maximizes mIoU. In NetDissects measure in contrast, the activation binarization threshold θ_k is chosen based on the top quantile level of activations $a_k \in A_k$ such that $P(a_k > \theta_k) = 0.005$. This fails for non-Gaussian distributions, e.g. Bernoulli, for which θ_k could wrongly be set to 1, resulting in M_k to be always 0.

Insensitivity to size of concept. To show size bias using IoU, we conduct a comparison between IoU and mIoU. We compare concept assignments on SSB@pyramid since the channels are pre-assigned. Table 2 presents the assignments of each method (columns) for four channels (rows). mIoU assignments are consistent with the trained concepts, even identifying concept *wood*. Using IoU instead, concepts like *painted*, or *black* are among the identified. These concepts cover large areas in Broden images making them less susceptible to noise. The average pixel portion per image of *painted* for example is 1087.5, resulting in an IoU of 0.06, while hair has only 93.8 pixels on average and does not show up when using IoU. mIoU on the other hand computes

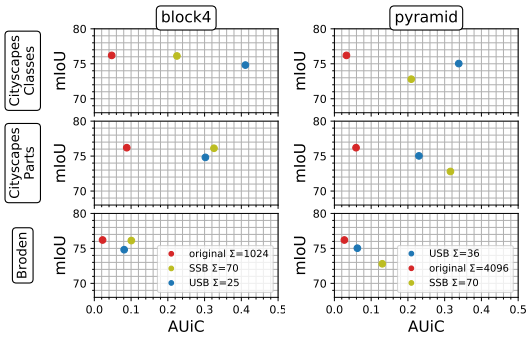

Figure 1: AUiC - inspectability scores for SSBs (yellow), USBs (blue) and baselines (red). Higher values are better, 1 being perfect channel-concept alignment. SBs substantially improve that alignment and thus: inspectability. Σ indicates number of channels.

a score for hair of 0.55 for channel 16, which is trained for *hair*. NetDissects metric uses IoU, for which the authors manually adjusted the threshold to make it unbiased [13]. Since this adjustment is done for normal distributions, it's not guaranteed to be unbiased for others.

5. Results

We here show, that introducing Semantic Bottlenecks (SBs) achieve all of our three goals (i)-(iii). To assess the semantic alignment of channels (goal (ii)) we utilize our AUiC metric to show improved inspectability for SBs over baseline layers. Additionally, we plot the mIoU performances showing we can recover performance, achieving goal (i) and (iii).

5.1. Setup

Datasets. We compare alignments with three different datasets to cover a wide range of concepts from different domains. The broadest dataset we evaluate is Broden [1] which covers 729 various concepts of categories like object, part, material, texture and color (skipping scene concepts). Since the Broden images are mostly out of domain w.r.t. Cityscapes, we evaluate *Cityscapes-Classes* and *Cityscapes-Parts*, a dataset we introduce to include subordinate concepts to the 19 classes. The new dataset includes 11 coarsely annotated images covering 38 different concepts.

Compared models. Given 70 channels for SSBs, we choose 25 and 36 channels for USBs.

5.2. Quantitative improvements

We compare vanilla PSPNet with SSBs and USBs and do so for outputs of block4 and the pyramid layer. The plots for these two layers are presented in columns in figure 1. Each row shows results for a different dataset in this order: Cityscapes-Classes, Cityscapes-Parts and Broden. PSPNet layer outputs are indicated by color *red*, SSBs by *yellow* and USBs by *blue*.

SSBs enable inspection for subordinate concepts. On each layer and dataset except Cityscapes-Classes, SSBs out-

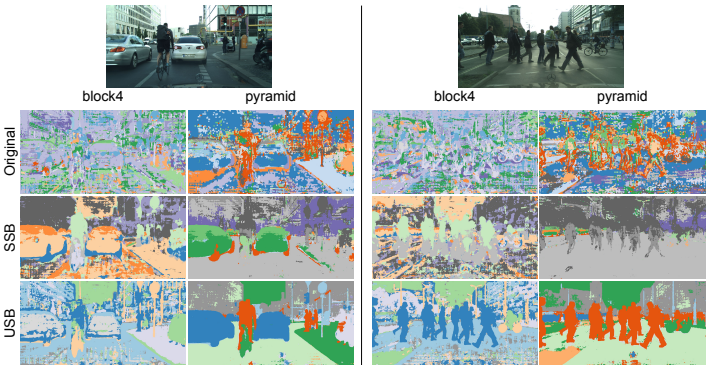

Figure 2: Top-20 Broden aligned channels from SSB-, USB- and vanilla PSPNet outputs. Each color is mapped to a single output channel.

perform baselines. Most encouragingly, SSBs improve the AUiCs on Cityscapes-Parts from under 0.1 to over 0.3 for both block4 and pyramid making a big leap forward towards inspectable representations.

USBs delineate Cityscapes-Classes related concepts. In comparison to SSBs, USBs align very well with Cityscapes-Classes. The increase from 0.05 to over 0.4 AUiC on block4 is especially remarkable.

5.3. Qualitative improvements

To support our quantitative results we supply visualizations of SB-layers in comparison to baselines. We show that SB outputs offer substantially improved spatial coherency and consistency. To enable comparison between 1000s and 10s of channels, we utilize the mIoU scoring of our metric to rank channels. We show the top-20 channels, assigning each a unique color and plotting the $\arg\max$ per location. Based on our discussion of inspectable channels, this will result in coherent activations for unique concepts *if a channel is aligned*. Visualizations are presented in figure 2 for all tested layer locations.

PSPNet outputs in the first row (*Vanilla*) show that they are very difficult to inspect even if ranked by best aligned channels. This leads us to believe that representations are highly distributed across channels.

SSB outputs. Attending to the first image on the left half of figure 2, we see spatial coherency greatly improved for SSB and USB outputs over baseline. In particular, note the responses for SSB@block4 which show a distinction into wheels (blue color), car windows (dark orange color) and person-legs (light gray color). A similar distinction for SSB@block4 can be seen for the second input image, where there is one channel activated for the upper body (mint green) and one for the legs (light gray). We find that our SB-layers offer dramatically increased inspectability that give insights into strong correlations between channel output and distinct input features.

USB outputs. In relation to the SSB outputs, the USBs appear to form representations that are early aligned with the output classes, which is especially evident for USB@pyramid where the visual distinction between classes is already very sharp.

6. Conclusion

We proposed supervised and unsupervised Semantic Bottlenecks (SBs) to align deep representations with semantic concepts. Additionally, we introduced the AUiC metric quantifying such alignment to enable model agnostic benchmarking and showed that SBs improve baseline scores up to four fold while retaining performance.

References

[1] Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: CVPR (2017) 1, 2, 3

[2] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 2

[3] Li, L.J., Su, H., Fei-Fei, L., Xing, E.P.: Object bank: A high-level image representation for scene classification & semantic feature sparsification. In: NIPS (2010) 1

[4] Lin, D., Shen, X., Lu, C., Jia, J.: Deep lac: Deep localization, alignment and classification for fine-grained recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1666–1674 (2015) 1

[5] Lipton, Z.C.: The mythos of model interpretability. Queue **16**(3), 30 (2018) 1

[6] Melis, D.A., Jaakkola, T.: Towards robust interpretability with self-explaining neural networks. In: NIPS (2018) 1

[7] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., et al.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV. pp. 618–626 (2017) 1

[8] Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: ICML (2017) 1

[9] Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV (2018) 2

[10] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization. arXiv:1506.06579 (2015) 1

[11] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014) 1

[12] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017) 2

[13] Zhou, B., Bau, D., Oliva, A., Torralba, A.: Interpreting Deep Visual Representations via Network Dissection. arXiv e-prints arXiv:1711.05611 (Nov 2017) 3

[14] Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing deep neural network decisions: Prediction difference analysis. arXiv:1702.04595 (2017) 1