Examining Interpretable Feature Relationships in Deep Networks for Action recognition

Anonymous Authors[1]

Abstract

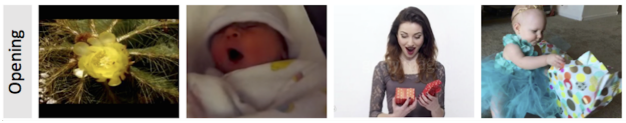
A number of recent methods to understand neural networks have focused on quantifying the role of individual features. One such method, NetDissect identifies interpretable features of a model using the Broden dataset of visual semantic labels (colors, materials, textures, objects and scenes). Given the recent rise of a number of action recognition datasets, we propose extending the Broden dataset to include actions to better analyze learned action models. We describe the annotation process, results from interpreting action recognition models on the extended Broden dataset and examine interpretable feature paths to help us understand the conceptual hierarchy used to classify an action.

1. Introduction

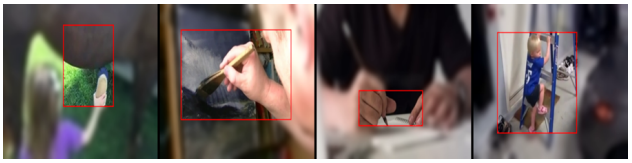
The success of Deep convolutional neural networks (DNNs) is partly due to their ability to learn hidden representations that capture the important factors of variation in the data. Previous works have visualized the units of deep convolutional networks by sampling image patches that maximize the activation of each feature (Zhou et al., 2016a) or by generating images that maximize each feature activation. Such visualizations show that individual features act as visual concept detectors. Features at lower layers detect concrete patterns such as textures or shapes while features at higher layers detect more semantically meaningful concepts such as dog heads or bicycle wheels. One tool for network interpretability (NetDissect) (Bau et al., 2017; Zhou et al., 2018) uses the Broden dataset (consists of objects, scenes, object parts, textures and materials) to evaluate individual units.

Recently, DNNs have shown significant progress in action recognition with the introduction of large-scale video datasets. However, while NetDissect with the Broden dataset is appropriate for networks trained on object or scene recognition, it does not include the ability to detect learned action concepts.

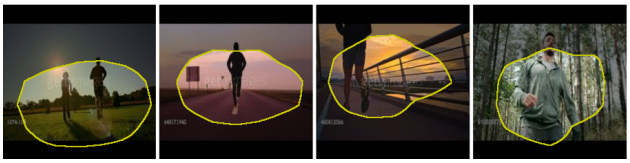
In this paper, we propose extending the Broden dataset to include actions so that we can more appropriately interpret action recognition networks. We describe our annotation process to collect images across action classes and select regions of importance for identifying each action. We then show results using our Action Region dataset together with the existing Broden set to identify interpretable action features in deep networks trained for action recognition. The Action Region dataset presented, and the code for integrating with NetDissect, will be made available online.



Sample videos Example frames from a few videos to show intra-class action variation



Action Regions Spatial localization of actions in single frames for network interpretation



Action Interpretation Identifying interpretable action features

2. Identifying Action Features

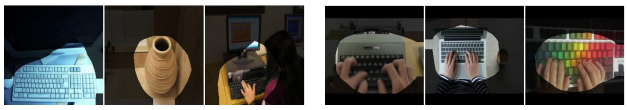
To better analyze action models, we extend the Broden dataset to include actions. This is done by first building an image segmentation dataset for actions.

2.1. Annotation

We begin by collecting bounding box annotations via Amazon Mechanical Turk (AMT) for actions in images selected from videos, for which we use the Moments in time dataset (Monfort et al., 2019). We extract a single frame from the center of 500 randomly selected videos for each of the 339 action classes from the dataset and present a binary annotation task to the workers on AMT asking if an action from the source videos label set is visible in the frame shown. This binary interface is very similar to that used for collecting the action labels for the Moments in Time dataset (Monfort et al., 2019) with the main difference being the

use of images rather than video. We run this task for at least 2 rounds of annotation to verify that the action is visible in each frame. We then take the set of verified action-frame pairs and pass them to a separate annotation interface on AMT that asks the workers to select the regions most important in the image for identifying the action. Multiple regions can be selected for an image as in the jogging example in Figure 3 and the workers are allowed to skip an image if there are no useful regions for detecting the action (i.e. the action is not visible in the image).

We run this region selection task through multiple rounds and only consider overlapping regions from the different rounds as most important for detecting the actions. After this stage the regions selected are cropped from the original images and passed through the binary annotation task previously described for a final verification that the actions are present and recognizable in the selected regions. After our complete annotation process our total set of verified images with segmented action regions consists of 23,244 images from 210 different classes. Figure 3 displays some examples of the selected regions collected through this process.



(a) Cracked (IoU 0.1) (b) Typing (IoU 0.34)

Figure 2: Example of the same feature (290) evaluated using the (a) original Broden dataset and (b) the proposed Broden+Action dataset.

2.2. Action region dataset

To integrate our new action region dataset into the NetDissect framework, we first consider each selected region to be a mask on the segmented area of the image relating to the action. This is similar to part, material and object masks used for other segmentation datasets (Zhou et al., 2016b; Chen et al., 2014; Mottaghi et al., 2014; Bell et al., 2013). With the data formatted in this manner we extend the Broden dataset to include our action segmentations and extract the set of interpretable action features detected via NetDissect. This process allows us to identify not just object, scene, texture and color concepts learned by our models, but actions as well. In Section 3 we show some of the key results from interpreting action networks in this way.

3. Experiments

To score and quantify the unit interpretability of a network we follow the same procedure as outlined in (Bau et al., 2017). All experiments use a ResNet50 network (He et al., 2016) trained on the Moments in time dataset (Monfort et al., 2019) for classification performance. We analyze features from the outputs of the residual blocks (referred to as block1, block2, block3 and block4 corresponding to conv2, conv3, conv4 and conv5) of the network.

Category	Concepts	Interpretable Features
Broden	108	850
Action Regions	141	1971
Broden+Action Regions	193	1978

Table 1: Comparison of the number of concepts and interpretable features identified by NetDissect given the Broden dataset, the Action Region dataset and the combined dataset on block 4 of a ResNet50 trained for action recognition.

3.1. Action Dissection

Using the approach described in Section 2 we are able to identify 141 action concepts learned in 1971 different features out of 2048 (Figure 4) units in the final convolutional layer (block4) of a Resnet50 network trained on the Moments in Time dataset. Figure 5 highlights some of the learned concepts. Interestingly the network seems to be recognizing the pattern of a person standing behind a podium as *preaching* which is definitely a common correlation in our dataset. Similarly, the network associates *crawling* with babies as many of our videos of crawling typically depict babies *crawling*. These are the types of data and class biases that are useful to identify via network interpretation that may have gone unnoticed without the ability to identify action concepts.

Table 1 highlights the fact that including actions in the Broden dataset helps to interpret a much larger portion of the features in block 4 of a ResNet50 trained for action recognition. With the original Broden set (no actions) NetDissect identified 108 concepts in 850/2048 features. Adding actions to the Broden set allowed us to identify 1978/2048 features that can be interpreted for 193 different concepts including 141 actions. This large jump in the number of interpretable features makes sense for the final block of a model trained for action classification and suggests that excluding action concepts misses a large amount of useful information each feature represents. Combining the original Broden set with the proposed Action Regions results in identifying a much larger number of concepts, 193 concepts in 1978/2048 features (96.5% of the features). The results from the combined set highlight that some of the features previously interpreted by the original Broden set as object or texture concepts are closely aligned with actions. For example, unit 13 was classified using the Broden set as learning the concept "potted plant" with an IoU of 0.06, but if we include action concepts the unit is found to be more correlated with the action "gardening" with an IoU of 0.15. Similarly, unit 290 was identified by Broden as learning the texture concept "cracked" with an IoU of 0.1 and including actions we found a greater association with the action "typing" with

bicycling floating grooming writing jogging

Figure 3: **Visualization of labelled regions**

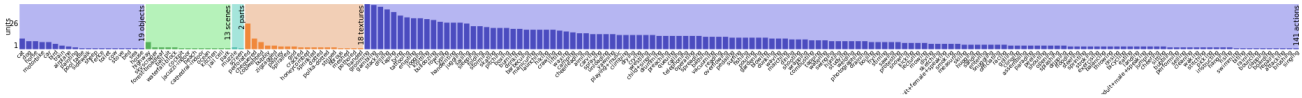


Figure 4: **Graph of learned concepts ordered by the number of features associated with each concept.**

Figure 5: **Visualization of different features that are interpretated as learning the same action concept. Many features that share the same action interpretation seem to learn different representations of the same action.**

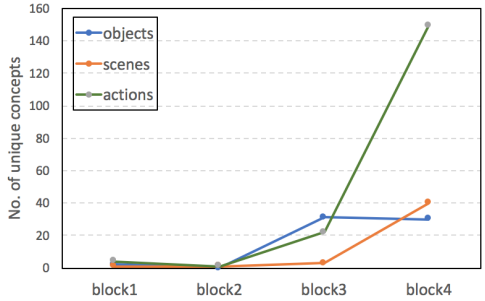


Figure 6: **ResNet block-wise interpretability** Visualize how different semantic concepts - objects, scenes and actions emerge across residual blocks of the ResNet50 network.

an IoU of 0.34. Features for identifying the ridges between the keys in the keyboards commonly found in actions of "typing" were correctly activating for the texture "cracked", however we can see from Figure 2 that the feature is more correlated with the action "typing".

3.2. Block-wise Interpretability

To understand how individual units evolve over residual blocks we evaluate the interpretability of features from different blocks of a resnet50 network trained for action recognition on the Moments in Time dataset (Monfort et al., 2019) in terms of objects, scenes, actions and textures. We observe that action features mainly emerge in the last convolutional block (block 4) of the model. It is interesting to note that object and scene features are learned even if the model is not explicitly trained to recognize objects or scenes suggesting that object and scene recognition aids action classification.

3.3. Interpretable feature relationships

Examining these interpretable features and how they contribute to a networks output allows us to build a better understanding of how our models will function when presented with different data. For example, we can consider a fea-

Figure 7: **Example activation of a feature interpreted to have learned the scene "highway".**

(a) (b)

Figure 8: An image incorrectly classified as "juggling" (a) with the CAM of the model showing a stong activation on the purse (b).

ture in the final residual block of the network that has been interpreted as a "highway" scene feature (Figure 7. If we activate only this unit by setting its values to 1 and all other feature (including bias) values to 0 and examine the output we can identify which action classes are using the fact that a video may take place on a "highway". In this case the action classes that achieve the highest output are "hitchhiking", "towing", "swerving", "riding", and "driving". These interpretable feature-class relationships make sense as all of these actions are likely to occur near a "highway".

Similarly, we can examine the relationships between different interpretable features at different layers in the network to help us understand the concept hierarchy the model is using to make a decision. Understanding this process can be very useful in diagnosing why a network makes a mistakes in its output. For example, if we pass the image in Figure 8a through our action recognition model we get a top prediction of "juggling". Of course the image does not depict the action "juggling". To diagnose why the network was incorrect we performed the inverse of the operation described in the previous paragraph and iteratively set the value of each feature in block4 of the model to 1, and the others to 0, compared the resulting outputs for the action "juggling" and identified the interpretable features that contributed the most to the mistake (i.e. resulted in the highest output for "juggling"). Unfortunately, in this case the top 5 features that contribute to the class "juggling" are also interpreted as "juggling" features and none of these features share a significant correlation with any other concept in the Broden dataset.

To address this we ran the same process again on block3 by considering the features in block4 as the output and compared the activations of the "juggling" features. This step was much more informative on how the model arrived at its mistake. We found that the interpretable concepts in block3 that that contributed the most to the "juggling" features in block4 were the scene "ball pit", the textures "polka dotted" and "dotted" and the color "white". We can see that the woman in the image is holding a red and white polka-dotted purse and running Class Activation Mapping (CAM) (Zhou et al., 2016a) on the image confirms that the area around the purse is an area of interest for the model (Figure 8b). We were thus able to identify the hierarchical concept path within the network that led to the misclassification.

4. Conclusion

We introduced Action Regions to the Broden dataset to allow for NetDissect to identify action concepts learned by interpretable features in networks trained for action recognition. We showed the resulting increase in identifying interpretable features and learned concepts and highlighted some interesting examples. Future work will focus on expanding feature interpretation for spatio-temporal networks trained for video understanding.

References

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

Bell, S., Upchurch, P., Snavely, N., and Bala, K. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Trans. on Graphics (SIGGRAPH)*, 32(4), 2013.

Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., and Yuille, A. Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S. A., Yan, T., Brown, L., Fan, Q., Gutfruend, D., Vondrick, C., et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–8, 2019. ISSN 0162-8828. doi: 10.1109/TPAMI.2019.2901464.

Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Urtasun, R., and Yuille, A. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016a.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016b.

Zhou, B., Bau, D., Oliva, A., and Torralba, A. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.