02456 DEEP LEARNING, DTU COMPUTE, FALL 2022

BEEN KIM'S "QUANTITATIVE TESTING WITH CONCEPT ACTIVATION VECTORS (TCAV)": A CRITICAL REVIEW

Aldís H. Björgvinsdóttir, Bernharð A. Jónsson, Hildur M. Gunnarsdóttir, Pálína K. Guðmundsdóttir, Sara H. Jósepsdóttir

DTU Compute, Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark

ABSTRACT

Understanding deep learning models is a significant challenge due to their size, complexity and interpretability. Interpretability ensures that machine learning (ML) model predictions reflect human values. A paper on "Quantitative Testing with Concept Activation Vectors (TCAVs)" [1] introduced a linear interpretability method, quantitative TCAV, to address this challenge. TCAV's primary goal is to explain how humans communicate with one another [2], by reflecting on how essential a concept is for predicting the trained model. Another significant challenge is the reproducibility of results due to factors such as changes in data and hyperparameter inconsistency [3]. We addressed these challenges by critically reviewing the aforementioned paper and reproducing experiments presented in the paper on two classes, figures of fire engines and suits. Even though the results were similar for the TCAVs, the performance was nowhere near as high as in the original paper. This can mainly be attributed to the immense difference in number of experiments that were carried out. There were also inconsistencies between the information in the original paper and the code provided by the authors in addition to lack of information in general. These findings highlight the pervasive issue of reproducing experiments and demonstrate that interpretability remains a significant challenge in the field of ML.

Index Terms— Deep Learning, Concept Activation Vectors, Interpretability, Human Perception

1. INTRODUCTION

It remains a significant challenge to understand the behaviour of modern machine learning (ML) models. This is why the focus has shifted from solely improving the model's performance to improving its interpretability. In addition to prediction accuracy, interpretability is an essential factor to consider to ensure that the ML model reflects human values.

Interpretability can be thought of as describing the model's predictions utilizing its input features. For instance, for linear models such as logistic regression, one can get the importance of the weights of each feature. Other methods, such as saliency maps, give importance weights to each pixel, not responding to high-level concepts easily understood by humans.

Internal values of the ML model, that is, the neural activations, can appear incomprehensible to the human eye. The paper on "Quantitative Testing with Concept Activation Vectors (TCAV)" introduces a new linear interpretability method, TCAV, to address this problem. For a given concept (e.g., stripes when classifying images of zebras), the TCAV measures the extent of that concept's influence on the model's prediction for a specific class. The relationship between a concept and a class provides helpful insight into a model's reason for prediction and overall behaviour [1].

Another significant challenge in ML models is the reproducibility of results. This can be due to a lack of records, changes in data and hyperparameter inconsistency [3]. Decisions are often made "silently" through default parameters that a given software library has predetermined, making simple documentation of the exact configuration, which may involve millions of parameters, challenging in certain situations [4]. It is nearly impossible to reproduce models without using the same development environment and software frameworks used to generate the model [3]. In addition, numerous ML models, particularly deep learning models, use randomness in their training which leads to different parameter values being selected each time [4].

The primary goal of this report is to address these challenges by conducting a critical review of [1]. The results of this report are thus a comparison of the performance of the TCAV by reproducing experiments presented in the paper. The experiments involve computing a quantitative explanation by the use of TCAVs of the relative importance of the concepts related to each class.

2. METHODS

This section explains the ideas and methods used in the project. The implementations are based on codes in Tensorflow's *tcav* GitHub repository [2]. The code used for the experiments in this paper can be found in this GitHub repository [5]. The neural networks we consider are models with inputs $x \in \mathcal{R}$ and a feed-forward layer l with m neurons, where the function $f_l : \mathcal{R}^n \rightarrow \mathcal{R}^m$, represents the input inference and its l cordo.

2.1. Linear Interpretability

The challenge of understanding ML models can be expressed mathematically by looking at the model as a vector space E_m, spanned by basis vectors e_m, which correspond to neural network components. As one might expect, humans work in a different vector space E_h, spanned by e_h, corresponding to an unknown human-interpretable concept. This interpretation can be noted by the function:

$$g : E_m \rightarrow E_h \tag{1}$$

When g is linear, it's called linear interpretability. Concept Activation Vectors (CAVs) are used as a translation between these vector spaces, E_m and E_h.

2.2. Concept Activation Vectors (CAVs)

Given a set of examples representing a concept of human interest and a random set of examples without a concept. The CAV is a vector in the space of activations of layer l that represents this concept. It is learned by training a linear classifier to distinguish between the activations produced by a concept's examples and examples in any layer. The vector is then obtained by finding the normal to a hyperplane separating the examples in the model's activations (see red arrow in Figure 1).

To provide a mathematical explanation. Now defining a binary linear classifier, $v_c^l \in \mathcal{R}^m$, which is a linear CAV for the concept C. This classifier can be trained to distinguish between the layer activations of the two sets: $\{f_l(\mathbf{x}) : \mathbf{x} \in P_C\}$ and $\{f_l(\mathbf{x}) : \mathbf{x} \in N\}$, where P_C is a positive set of examples (e.g. photos of dotted objects), and N is a negative set of random examples (e.g. random photos).

2.3. Testing with CAVs (TCAV)

The "conceptual sensitivity" of class k to concept C can be computed as the directional derivative:

$$\begin{aligned} S_{C,k,l}(\mathbf{x}) &= \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(\mathbf{x}) + \epsilon v_c^l) - h_{l,k}(f_l(\mathbf{x}))}{\epsilon} \\ &= \nabla h_{l,k}(f_l(\mathbf{x}))| \cdot v_c^l, \end{aligned} \tag{2}$$

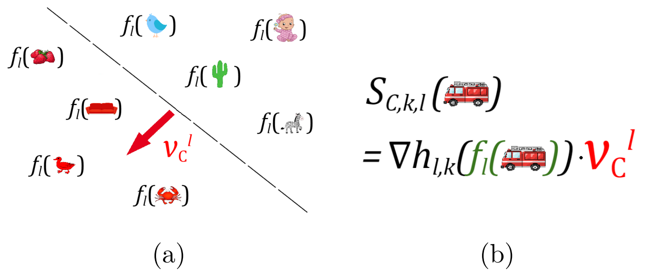


Fig. 1: (a) The CAV points to the direction of the concept, v_C^l (red arrow). (b) The directional derivative $S_{C,k,l}(x)$ is used by TCAV to quantify conceptual sensitivity. An example class of interest is the Fire Engine.

where $h_{l,k} : \mathcal{R}^m \rightarrow \mathcal{R}$ and $f_l(\mathbf{x})$ represents the activations for input $\mathbf{x}$ at layer l. Testing with CAVs (or TCAV) uses this directional derivative to compute the conceptual sensitivity of ML models across entire classes of inputs. By utilizing $S_{C,k,l}(\mathbf{x})$, we can quantitatively measure the sensitivity of model prediction concerning concepts at any model layer.

Given a supervised learning task and letting X_k denote all inputs with a given class label k, we define the TCAV score as the fraction of k-class inputs whose l-layer activation vector was influenced positively by concept C:

$$TCAV_{Q_{C,k,l}} = \frac{|\{\mathbf{x} \in X_k : S_{C,k,k}(\mathbf{x}) > 0\}|}{|X_k|}, \tag{3}$$

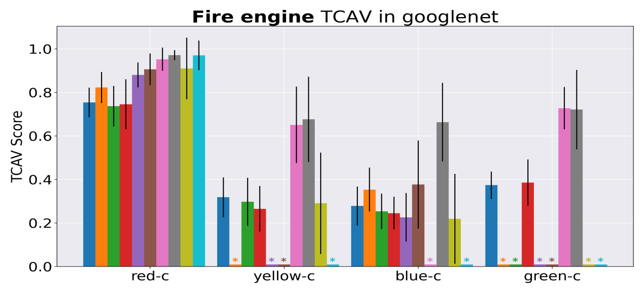
where $TCAV_{Q_{C,k,l}} \in [0, 1]$. This metric allows us to interpret the conceptual sensitivity easily.

2.4. Relative TCAV

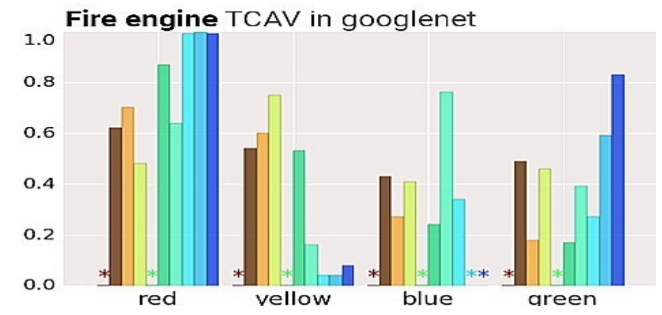
In practice, CAVs are often far from orthogonal due to closely related concepts, such as similar colors and patterns. This applies to image recognition as we can hypothesize that these concepts are likely to exist as internal representations. Given three positive sets of similar colors, that is different shades of red. A relative CAV can be derived by constructing, for each example concept, a negative input set by complement, e.g., $\{P_{red} \cup P_{pink}\}$ for the color *cordovan*. The $TCAV_Q$ score in the following experiments was therefore computed using relative CAVs [1].

2.5. Statistical Testing

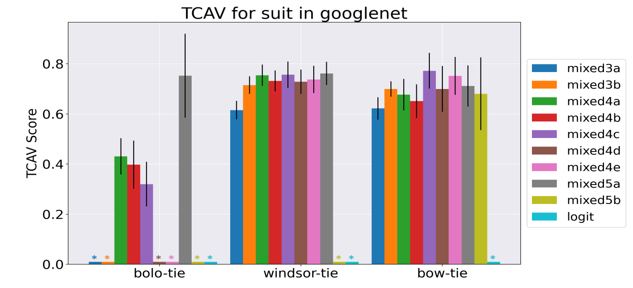
To avoid producing meaningless CAVs, statistical tests were performed. This is an important step since CAVs will be produced every time with TCAV, even if the data, and as a result the concept, is completely random. To prevent this, each CAV is trained against multiple sets



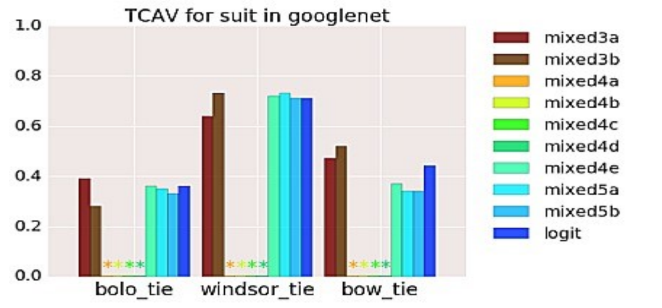
(a) Fire Engine experiment.



(b) Fire Engine experiment from the original paper [1].



(c) Suit experiment.



(d) Suit experiment from the original paper [1].

Fig. 2: Relative TCAVs for individual layers in GoogLeNet [6]. CAVs omitted after statistical testing are marked by $*$. The concept **red** for the Fire Engines (a) had the highest importance, with similar results as the original paper (b). There are more dissimilarities in the results for the Suit experiment. The concepts **windsor tie** and **bow tie** for Suits have the highest importance in (c) but it seems **windsor tie** is deemed most important in the original paper (d).

of random examples. A truly meaningful concept is then expected to produce a consistent TCAV score throughout these random training runs [1]. The original paper mentions performing 500 random runs for each CAV. However, due to computational limitations, only 25 random runs were performed for each CAV in this experiment. Each random run was performed for 200 example images.

A two-sided t-test was performed on the TCAV scores to determine the significance of the results, examining if the hypothesis of a TCAV score equal to 0.5 could be rejected. Additionally, a Bonferroni correction was performed for our hypothesis (at $p < \alpha/m$ with $m = 2$) corresponding to what was mentioned in the original paper.

2.6. Obtaining Concepts for Testing

The concept of interest is defined by choosing a set of examples where the concepts for the Suit experiment were gathered from ImageNet dataset [7], but the concepts for the Fire Engine experiment were gathered from the Broden dataset [8].

The method of extracting color concepts was developed based on information provided by *Bau et al* regarding image color data [9]. The color concept data was obtained using image data from the Broden dataset [8]. Each image in the dataset was assigned a color based on its most frequently occurring common color, using the 11 common human perceptible colors defined by *van de Weijer* [10]. A random sub-sample of 200 images for each color of interest was then extracted from the dataset to be used in the TCAV.

2.7. Hyperparameter Tuning

To accurately reproduce the results, hyperparameter tuning needed to be performed. Two main parameters required tuning, the hyperparameter α, used to compute the learning rate, and the amount of random data used in the statistical testing. The original paper clearly states the amount of random data recommended for the experiments but does not mention the α parameter. Hyperparameter tuning of the α parameter was performed by running the TCAV with 10 different values for α, see appendix A, one of which was the value defined in the code obtained from GitHub [2].

3. RESULTS

Both the results from the recreation of the experiments and the experiments from the original report can be seen in Figure 2, with the recreated results on the left-hand side and the results from the original paper on the right-hand side.

For the Fire Engine experiment, the results are quite similar to the original report, where the **red** concept has the highest importance by far. On the other hand, the **yellow** concept shows much less importance than in the original report. The other two colors, **blue** and **green** show a similar magnitude of importance. For all of the concepts, there is a difference in the bottlenecks that were deemed insignificant. For the Suits experiment, the **windsor tie** and **bow tie** concepts show the highest importance, with **bolo tie** being much less significant. The original report shows the **windsor tie** with the highest importance, but with **bow tie** and **bolo tie** similar. There is also quite a difference in which layers are deemed insignificant, this is especially true for the Suit experiment.

Regarding the performance of the TCAVs in learning concepts, the mean accuracy of the linear classifiers used to train the CAVs for each layer in GoogLeNet was compared to the one in the original paper. This was done for both the Suit and Fire Engine classes. Figure 3 shows that the TCAV performances are nowhere near as high as in the original paper with accuracies between only **0.53** and **0.56**. In addition, in the original paper simple concepts such as colors achieved higher performance in lower-layers than more complex concepts such as objects. This is also true for the reproduced experiments, even though the difference is slight and the performance is more even throughout the layers.

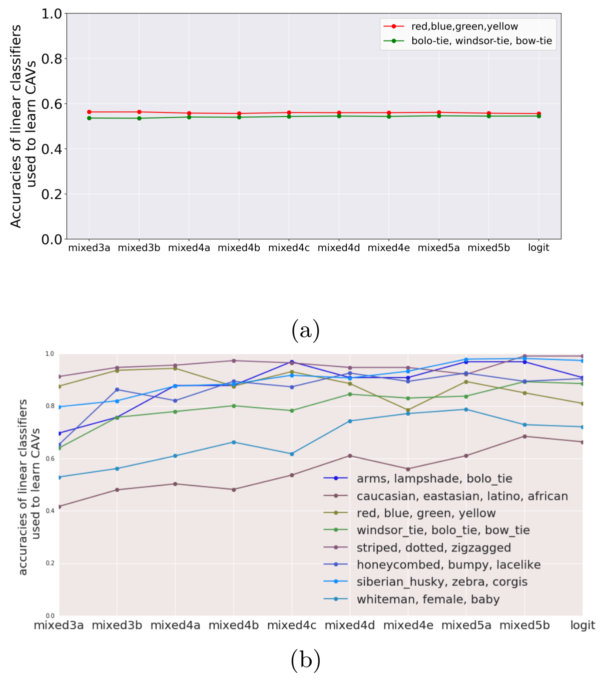


Fig. 3: Accuracies of linear classifiers used to train the CAVs for each GoogLeNet layer. The TCAV performances are nowhere near as high in (a) as in the original paper (b).

4. DISCUSSION

The TCAV results from the experiments show that the concepts **red** for Fire Engines as well as **windsor tie** and **bow tie** for Suits have the highest importance. This is fairly consistent with the results from the original paper. However, the distribution of the concepts' influence in the layers varies quite a bit from the original. Furthermore, the results revealed that the TCAVs performance in learning concepts varied significantly, with the original paper having much higher accuracies. Nonetheless, we cannot be sure how the accuracy was computed in the paper, as it was not mentioned. As a result, we presumed it was the average accuracy per layer for all experiments.

The discrepancies between the results in the original paper and the results produced in this experiment can largely be attributed to a number of difficulties in recreating the experiments. Information was lacking in the paper regarding exactly how the experiments were performed. For instance, the value of the hyperparameter used to compute the learning rate (α) was not mentioned in the original paper. We, therefore, performed hyperparameter tuning to select the value for α but cannot know for certain if there are differences between our value and the original. Another difficult parameter to match was the amount of data used. The original paper used 500 random experiments but due to a lack of computational capacity, only 25 experiments were carried out.

Additionally, there were discrepancies between the article and the code provided in the GitHub repository [2]. Although it is mentioned in the paper that Bonferroni correction was utilized to control the false discovery rate, it is not implemented in the code. Furthermore, attempting to recreate another experiment, the Dogsled experiment, which used TCAV on the InceptionV3 network instead of the GoogLeNet network, proved impossible because the InceptionV3 model's *.pb* file was not included in the GitHub repository.

Another issue could be that the editors are still actively editing the code in the GitHub repository, making reproducing the desired results more difficult than anticipated. Lastly, as was mentioned in the introduction, randomness can have a large impact on the results of the

ML model. When training the CAVs from the activations, stochastic gradient descent is used. Additionally, the GoogLeNet network uses a dropout layer [6]. This introduces randomness in the model, which can significantly affect the results. However, it is worth noting that randomness is an important element in deep learning as it facilitates the development of a generalized model.

5. CONCLUSION

To conclude, the main difficulty observed when recreating the experiment was the absence of certain critical information in the original report. The results from the recreation of the Suit and Fire Engine experiments are visually similar to the respective experiments in the original paper. However, the accuracies of the two experiments may be improved by increasing the number of random counterexamples, that is incorporating more data for training the CAVs, given the computational resources are available.

6. REFERENCES

[1] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al., "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*. PMLR, 2018, pp. 2668–2677.

[2] Been Kim, "tcav/tcav at master tensorflow/tcav," `https://github.com/tensorflow/tcav"`.

[3] Ejiro Onose, "How to solve reproducibility in ml," `https://neptune.ai/blog/how-to-solve-reproducibility-in-ml`[Accessed: 28-11-2022].

[4] Andrew Beam, Arjun Manrai, and Marzyeh Ghassemi, "Challenges to the reproducibility of machine learning models in health care," *JAMA*, vol. 323, 01 2020.

[5] Bernhard A. Jónsson et al., "Interpretability-using-TCAV/experiments at main bernhardjonsson/Interpretability-using-TCAV," `https://github.com/bernhardjonsson/Interpretability-using-TCAV"`.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

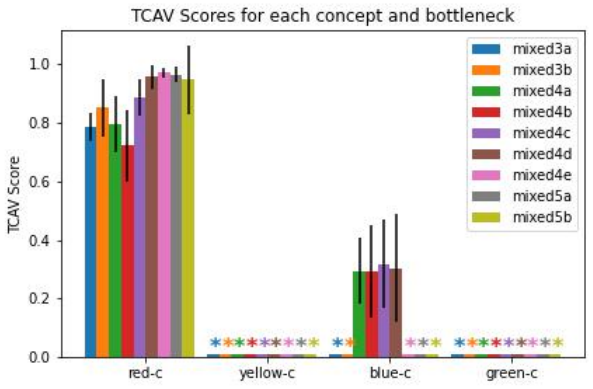
[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[8] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3319–3327.

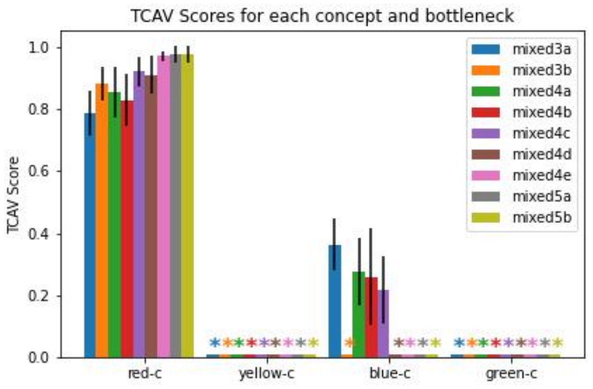
[9] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] Joost van de Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

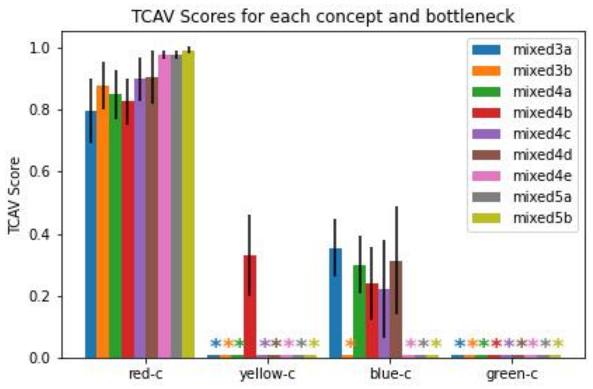
A. HYPERPARAMETER TUNING



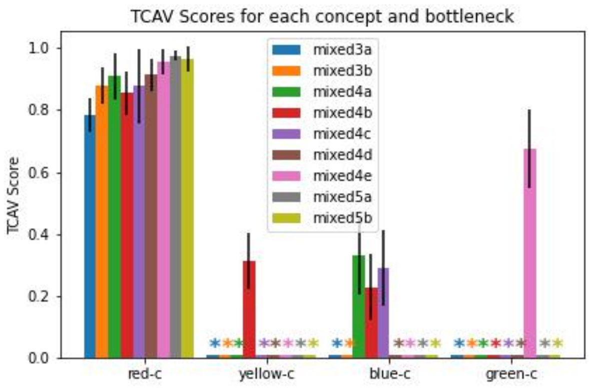
(a) Fire Engine experiment using $\alpha = 0.1$ and 8 random folders.



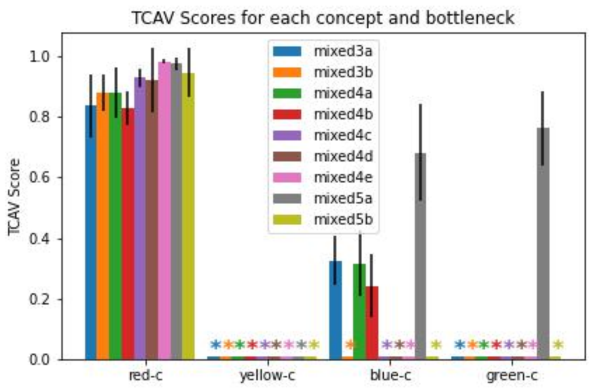
(b) Fire Engine experiment using $\alpha = 0.2$ and 8 random folders.



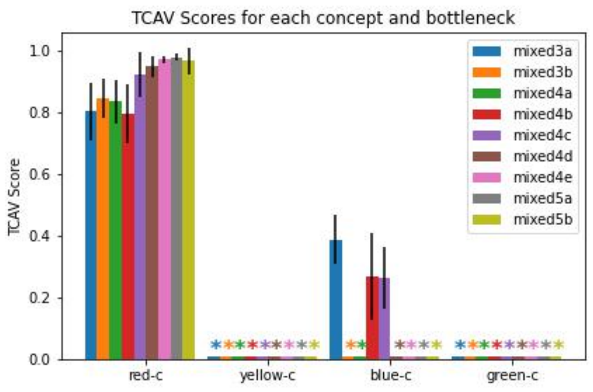
(a) Fire Engine experiment using $\alpha = 0.3$ and 8 random folders.



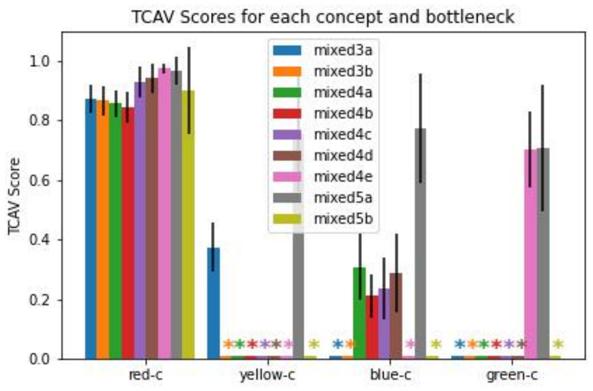
(b) Fire Engine experiment using $\alpha = 0.4$ and 8 random folders.



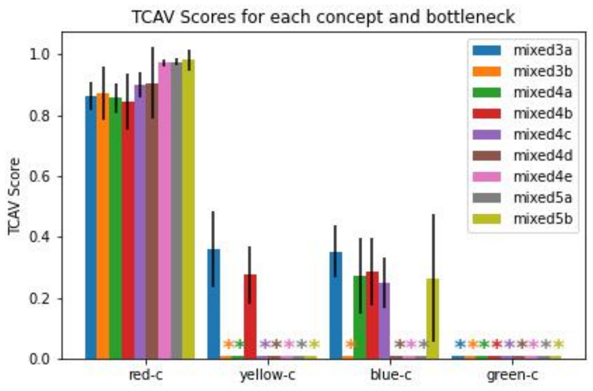
(a) Fire Engine experiment using $\alpha = 0.5$ and 8 random folders.



(b) Fire Engine experiment using $\alpha = 0.6$ and 8 random folders.



(a) Fire Engine experiment using $\alpha = 0.7$ and 8 random folders.



(b) Fire engine example using $\alpha = 0.8$ and 8 random folders.

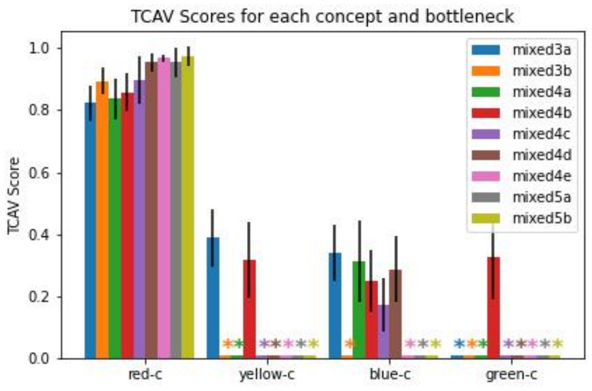


Fig. 8: Fire engine example using $\alpha = 0.9$ and 8 random folders.