



# Clustering for Explainability: Extracting and Visualising Concepts from Activation

Alexandre Lambert<sup>1,2,3</sup><sup>a</sup>, Aakash Soni<sup>1</sup><sup>b</sup>, Assia Soukane<sup>1</sup>, Amar Ramdane Cherif<sup>2</sup>  
and Arnaud Rabat<sup>3</sup>

<sup>1</sup>LyRIDS, ECE Research Center Paris, France

<sup>2</sup>LISV Laboratory, Université de Versailles, Paris Saclay, Velizy, France

<sup>3</sup>Unité d'Ergonomie Cognitive des Situations Opérationnelles, IRBA, Brétigny sur Orge, France  
{alambert, aakash.soni}@ece.fr

Keywords: Activations Explainability, Concept Extraction and Visualization, Clustering.

Abstract: Despite significant advances in computer vision with deep learning models (e.g. classification, detection, and segmentation), these models remain complex, making it challenging to assess their reliability, interpretability, and consistency under diverse. There is growing interest in methods for extracting human-understandable concepts from these models, but significant challenges persist. These challenges include difficulties in extracting concepts relevant to both model parameters and inference while ensuring the concepts are meaningful to individuals with varying expertise levels without requiring a panel of evaluators to validate the extracted concepts. To tackle these challenges, we propose concept extraction by clustering activations. Activations represent a model's internal state based on its training, and can be grouped to represent learned concepts. We propose two clustering methods for concept extraction, a metric for evaluating their importance, and a concept visualization technique for concept interpretation. This approach can help identify biases in models and datasets.


## 1 INTRODUCTION


Deep neural networks (DNNs) and convolutional neural networks (CNNs) are crucial for artificial intelligence thanks to their widespread availability and impressive performance on standardised benchmarks, particularly in computer vision applications. However, these models are often considered "black boxes", leaving users uncertain about their decision-making process and the knowledge they acquire. This lack of transparency make them less suitable for applications where interpretability is critical, such as medical diagnosis, autonomous driving, and human-centred models (Lambert et al., 2024). Thus, it is crucial to develop simple explanation methods to understand these models. Moreover, the explanation methods can provide several advantages. Firstly, they can provide enhanced model comprehension, allowing to interpret the model's inner workings, understand how it arrives at its predictions, and build trust in the model's decision-making process through better evaluation and refinement. Secondly, they can of-

fer valuable guidance during the training process and ensure that the model learns the desired information and avoid potential biases, leading to more robust and accurate model. Finally, these methods can help better understand outliers. In essence, these tools can offer a powerful perspective allowing non-specialists to gain deeper insights into the intricate world of DNNs and CNNs, enabling their use in various applications (Sivanandan and Jayakumari, 2020; Zhang et al., 2022; Atakishiyev et al., 2024) The state-of-the-art explanation methods are divided into two categories:

**Interpretable model** are neural network models designed to be inherently interpretable. They often incorporate human-interpretable concepts by training on custom loss functions and adding semantic knowledge into the networks (Wickramanayake et al., 2021).

**Post hoc explanations** methods can be applied to any model after it has been trained. These methods analyze the model's predictions and identify the most important features for those predictions. It is done by using feature maps, gradients or input perturbation. Post-hoc explanations can provide visual insights into

<sup>a</sup> <https://orcid.org/0000-0001-5702-6445>

<sup>b</sup> <https://orcid.org/0000-0002-0882-5280>

the model’s decision-making process and identify potential biases in the model (Lapuschkin et al., 2019).

This paper focuses on post-hoc explanations, particularly through analyzing activations. While the activation matrix shows the neural network’s internal state, it may not reveal the conceptual structures meaningful to humans or that the model is learning. To address this, we introduce a method to identify and group informative subsets of activations, referred to as concepts. Our method aims to make these extracted concepts interpretable and to assess their importance in relation to the model’s predictions. This paper proposes: 1) A method to extract concepts that highlight input image regions prioritized by the model for predictions. 2) A metric to assess the importance of these concepts. 3) A technique to visualize these concepts on the input image. Additionally, our code is publicly available to support the development of use cases.

The paper is organised as follows: Section 2 reviews related works. Section 3 presents our methodologies and two clustering methods for concept extraction. Section 4 presents the concept extraction results, followed by a discussion. Finally, the paper concludes in Section 5.

## 2 RELATED WORKS

Among post-hoc explainability techniques, attribution methods are widely used to determine the input variables contributing to a model’s prediction by generating importance maps. The Saliency method (Simonyan et al., 2014) creates heatmaps based on gradients to highlight influential pixels. GradCAM (Selvaraju et al., 2016) method incorporates gradients into class activation mapping. However, gradient-based methods can be limited because they capture model behavior in only a small local area around the input, potentially leading to misleading importance estimates (Ghalebikesabi et al., 2021). This is particularly true for large vision models, where gradients are often noisy and unreliable (Smilkov et al., 2017). To address this, perturbation-based methods, like Rise (Petsiuk et al., 2018), offer a valuable approach to understanding “where” a model focuses its attention, though they may be prone to confirmation bias, potentially leading to misleading explanations. This has led to questions about their usefulness. The HIVE framework (Kim et al., 2022), offers a way to assess explanations in AI-assisted decision-making scenarios, enabling falsifiable hypothesis testing, cross-method comparison, and human-centred evaluation of visual interpretability methods.

Recent approaches like ACE (Ghorbani et al.,

2019) focus on concept extraction by segmenting images and analyzing neural network activations, clustering them into “concepts.” However, ACE can include irrelevant background segments, necessitating post-processing to remove outliers. The ICE framework (Zhang et al., 2021) improves upon ACE by using Non-Negative Matrix Factorization (NMF) for better interpretability and fidelity, offering both local and global concept-level explanations. Similarly, CRAFT (Fel et al., 2023) employs NMF to extract concepts from model activations, refining them through recursive decomposition. However, CRAFT is more suited for groups of images and its methods for concept localization are complex, potentially challenging for non-experts.

To enhance interpretability, we propose a method that avoids the complexity of existing approaches, which often rely on “banks of coefficients” and computationally intensive steps that may obscure understanding at the single-image level. Our methodology to extract concepts uses less complex algorithm, maintaining efficiency and clarity, and making it more accessible to a broader audience.

## 3 METHODOLOGY

### 3.1 Overview of the Method

In this work, we investigate a supervised learning scenario, involving a pre-trained black box predictor  $M : X \rightarrow Y$  with a set of  $n$  images  $X \in \{x_1, \dots, x_n\}$  and their corresponding labels  $Y \in \{y_1, \dots, y_n\}$ . The input images are represented as a  $Ch \times H \times W$  matrix, where  $Ch$  represents the number of channels (e.g. RGB, RGBA, LA), and  $H$  and  $W$  are the image height and width. For each input image  $x$ , the predictor outputs  $M(x)$ . We assume that  $M$  is a neural network with fixed settings that can be divided into two parts:  $g$  transforms the input image into an intermediate representation  $g(x)$ , and  $h$  takes this intermediate representation to produce the final output  $M(x) = h(g(x))$ .

The intermediate representation is in a lower-dimensional space, determined by the number and nature of operations in  $g$  (e.g. convolution, pooling, down-sampling and scaling). For a given input  $x$ ,  $g(x)$  produces a set of activations  $\mathcal{A}$  with a shape  $A_N \times A_H \times A_W$ , where  $A_N$  is the number of activations, and  $A_H$ ,  $A_W$  are the height and width of each activation ( $A_i$ ).

In most pre-trained models, activations are typically non-negative due to the ReLU activation. The activation values within  $A_i$  can be viewed as a spatial distribution feature in a small information matrix.

Combining these values can help identify where specific information useful for classification is located. Essentially, when an image is passed through  $g(x)$ ,  $\mathcal{A}$  shows what the model has learned during training and where in the image it focuses during the forward pass, as these activation values are determinant for classification when fed into the classifier  $h(x)$ .

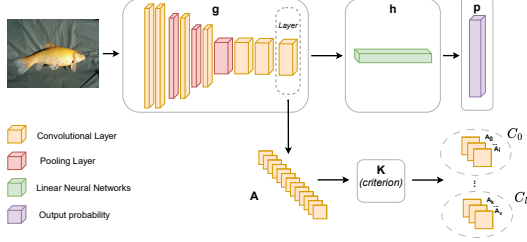


Figure 1: Method overview for concept extraction from a feature extractor  $g$  and a model classifier  $h$ . Any CNN architecture can replace  $g$  and  $h$ .

While the activation matrix comprehensively represents the neural network’s internal state, it may not directly reveal the underlying conceptual structures meaningful to humans. This motivates the exploration of methods to identify informative subsets of activations that can be grouped. We propose that when a sufficient number of  $A_i$  exhibits similar behaviour, they can be considered a set of cohesive units representing a learned concept  $C$ . Identifying these concepts helps gain insights into the model’s internal knowledge representation and facilitates a more nuanced understanding of the phenomena the model processes.

This work demonstrates that multiple activation patterns  $A_i$  can be grouped into different concepts  $C$  by satisfying specific criteria regarding a method  $K$ , as summarized in Figure 1. This approach aims to bridge the gap between raw activations and the high-level conceptual knowledge encoded by the model.

In the following paragraph, we propose two concept extraction methods: the first focuses on the internal patterns within each activation, and the second uses a relatively straightforward approach based on the position of high activation values.

## 3.2 Concept Extraction via Clustering

For a given  $\mathcal{A}$ , we aim to identify different concepts by regrouping different subsets of  $\mathcal{A}$  that satisfy a given criterion in a clustering method  $K$ . As mentioned earlier, the activation set  $\mathcal{A}$  is of shape  $A_N \times A_H \times A_W$ . However, to apply classical clustering algorithms without losing information, it is convenient to reshape  $\mathcal{A}$  as  $A_N \times (A_H \times A_W)$ , without any

need for normalisation.

The classical clustering algorithms require as input  $\mathcal{A}$  to produce a set of clusters  $\gamma = \{C_1, C_2, \dots, C_{N_{concept}}\}$  that exhibit the same clustering criterion. A concept  $C_l$  in  $\gamma$  obtained using a clustering algorithm  $K$ , is defined in Equation 1:

$$C_l = \{A_i \subseteq \mathcal{A} \mid f_K(A_i, C_l)\} \quad (1)$$

where  $f_K$  is minimised or maximised with respect to other clusters, depending on the algorithm  $K$ .

This work explores two possible ways of clustering to extract concepts, as explained in the following paragraphs.

### 3.2.1 Clustering Based on General Activations Patterns (CGAP)

This first approach focuses on obtaining concepts based on general activation patterns observed in  $\mathcal{A}$ . To achieve that, all the non-zero activations in  $\mathcal{A}$  are passed to the clustering algorithm  $K$ . A non-zero activation is  $A_i$  with at least one non-zero value. Since activations with all zero values do not play any role in classification, they can be ignored.  $K$  aims to regroup all the  $A_i$  that share similar activation values at similar indices, such that each  $C_l$  in  $\gamma$  contains unique sets of  $A_i$  from  $\mathcal{A}$ .

Given the high dimensionality of  $\mathcal{A}$ , applying Clustering directly to  $\mathcal{A}$  can be computationally intensive and may lead to sub-optimal clustering performance. As a solution, we employ Principal Component Analysis (PCA) as a dimensionality reduction technique before Clustering. PCA transforms the original high-dimensional activation data ( $A_H \times A_W$ ) into a lower-dimensional space while preserving as much variance as possible. This transformation helps highlight the most significant features contributing to the activation patterns, thus enhancing the effectiveness of the subsequent clustering process (Ding and He, 2004). The size of the lower dimension space depends on the number of desired concepts; in this study, it equals  $N_{concept} - 1$ . By reducing the number of dimensions, PCA helps enhancing computational efficiency and often improving the performance of clustering algorithms by emphasising the most distinctive clusters. After applying PCA, the reduced-dimensional activation data is fed into the clustering algorithm  $K$  to identify distinct activation patterns, extracting cohesive and informative concepts from the model’s learned representations.

It is important to note that the uniqueness of each cluster in  $\gamma$  can be evaluated and controlled using some metrics and criteria. However, the size of each cluster depends on the activation patterns, leading to

some clusters containing more activations than others, particularly in case of large activation patterns.

Depending on the application, if concepts representing small patterns in the input image are desired, the large clusters composing  $C_l$  can be divided into sub-clusters  $C_l^{sub}$  by iteratively applying the clustering algorithm  $K$  until the desired number of sub-concepts is extracted. In this work, the maximum number of sub-clusters is arbitrarily limited to 3.

### 3.2.2 Clustering Based on Position of High Activations (CPHA)

The second approach privileges regrouping activations  $A_i$  with higher values at similar spatial positions. Our observations suggest that high activation values often carry more weight in classification, as they correspond to the parts of the input image most relevant to the model's decision. Nevertheless, this may only sometimes be the case and warrants further investigation for generalisation. We propose that clustering activations with high values reveal concepts of relatively higher influence in classification and minimise redundancy in concept extraction. For that purpose, first, in each  $A_i$  the coordinates of  $\max(A_i)$ , called  $Coord_i$ , are identified as defined in the Equation 2

$$Coord_i = \arg \max A_i \quad (2)$$

Then, the clustering method  $K$  is applied on all the  $Coord_i$  to obtain  $\gamma$ . By using the set of  $Coord_i$  as clustering input, the concept extraction focuses on the spatial position of high activation values. Thus, concepts dispersed along the input image are identified, and the activations most relevant to the model's prediction are distinctly regrouped.

### 3.3 Concept Importance

To assess the importance of each  $C_l$  in classifying a target class (label), we propose a concept importance metric  $I_l$ , regardless of the concept extraction method.

For a given image  $x$  of target class  $t$ , first, we feed the model classifier  $h$  with  $\mathcal{A}$ . As output,  $h$  predicts the class  $t$  with a probability  $p_t$ . Then, to assess the importance of a concept  $C_l$  in the prediction of  $t$ , all the activation values of  $A_i$  in  $C_l$  are set to 0. The modified activation set is then fed to  $h$  to obtain a new prediction  $p_{c_l}$ . Finally, the importance  $I_l$  of concept  $C_l$  is then calculated from the difference between  $p_t$  and  $p_{c_l}$  as follows in Equation 3

$$I_l = \frac{p_t - p_{c_l}}{p_t} \times 100 \quad (3)$$

Note that, here, the concept importance is computed w.r.t a concept of interest  $C_l$ , and the sum of all concept importance is not equal to 100%.

Computing the importance of individual concepts provides valuable insights into how each concept contributes to the overall prediction score. A positive influence means that the given concept is responsible for a higher certainty of the model's prediction. In contrast, a negative influence makes the model's prediction less confident.

### 3.4 Concept Visualisation

Each concept  $C_l$  is a set of one or more activation  $A_i$  of shape  $A_H \times A_W$  (usually  $8 \times 8$ ), which is smaller than the input image shape (in our work, it is  $256 \times 256$ ). So, to project concepts onto the input image, an intermediate transformation is needed. It is achieved by, first, applying an element-wise sum among all the  $A_i$  in  $C_l$  and, then, interpolating the resulting matrix (of shape  $8 \times 8$ ) using bilinear interpolation to the input image size ( $256 \times 256$ ). The resulting matrix (of shape  $256 \times 256$ ) is finally min-max normalised. In the case of sub-clusters  $C_l^{sub}$ , the normalisation is performed using the minimum and maximum values of the parent concept  $C_l$  to ensure that the sub-concepts are visualised proportionally within the context of the overall concept.

## 4 RESULTS

A ResNet-50-based classification model pre-trained on the ImageNet-1k dataset is used to evaluate our concept extraction methods. The following paragraphs provide a brief description of the evaluation environment followed by a discussion on evaluation metrics and the result.

### 4.1 Evaluation Environment, Clustering Algorithms and Metrics

**Dataset:** ImageNet-1k (Deng et al., 2009) is a well-known extensive image database containing over a million images categorised into 1,000 different classes. We have arbitrarily chosen 11 classes for this study: rabbit (300 images), tench (387 images), english springer (395 images), cassette player (357 images), chain saw (386 images), church (409 images), french horn (394 images), garbage truck (389 images), gas pump (419 images), golf ball (399 images) and parachute (390 images).

**Model:** ResNet-50 (He et al., 2016) is a CNN architecture designed for image classification. It excels at identifying objects within images. thanks to its



deep architecture that learns complex patterns from the image. For the results presented in this paper, we use a pre-trained ResNet variant, called Norm-Free ResNet50 (Brock et al., 2021b; Brock et al., 2021a), that removes all normalization layers. The model has  $A_N = 2048$  activations in the last layer, each sized  $8 \times 8$ , and is initialized with ImageNet-1k weight configuration. The input image size is  $256 \times 256$ .

**Clustering Algorithms and Metrics:** We test our concept extraction method using four well-known clustering algorithms:  $k$ -means, Agglomerative, Birch and Gaussian Mixture Model (GMM). To evaluate the cluster quality representing the extracted concept, three metrics are used:

**Silhouette Score (SS)** measures the separation between clusters, with values range from -1 to 1. A score of 1 indicates well-separated clusters, 0 suggests overlapping clusters, and negative values indicate potential misassignments.

**Calinski-Harabasz Index (CHI)** (or Variance Ratio Criterion) evaluates between-cluster and within-cluster dispersion. Higher values indicate denser, more distinct clusters.

**Davies-Bouldin Index (DBI)** measures the average cluster 'similarity' by comparing inter-cluster distance with intra-cluster size. A lower index indicates better partitioning.

## 4.2 Evaluating Concepts Quality Based on the Clustering Metrics

In this section, we compare the performance of the clustering algorithms using the two methods (CGAP and CPHA) proposed in Section 3.2 for concept extraction. For comparison, the four clustering algorithms (Agglomerative, Birch, GMM and  $k$ -means) are used to extract  $N_{concept} = 5$  concepts from each input image (belonging to the 11 output labels) independently. The uniqueness and clustering consistency is assessed by comparing the clustering metrics for all the algorithms.

Table 1 shows the mean value of clustering metrics for different clustering algorithms using CGAP and CPHA methods. We observe that the  $k$ -means algorithm shows the best performance on all the metrics: 0.64 SS, 441.05 CHI and 0.97 DBI using CGPA, and 0.43 SS, 984.69 CHI and 0.84 DBI using CPHA. Both Agglomerative and Birch show similar or slightly lower performance than  $k$ -means. In contrast, GMM shows the worst performance. Additionally, the average execution time (in seconds) required for clustering for each algorithm is also compared in

Table 1, where Agglomerative is observed to be the fastest and GMM is the slowest.

Table 1: Comparison clustering method (mean over all labels).

Method	Cls	SS	CHI	DBI	Time
CGAP	A	0.61	398.70	1.00	0.14
	B	0.62	398.14	0.97	0.22
	G	-0.03	111.39	1.84	0.73
	k	<b>0.64</b>	<b>441.05</b>	<b>0.97</b>	0.20
CPHA	A	0.41	892.39	0.84	0.14
	B	0.37	724.83	0.90	0.22
	G	0.32	564.51	1.41	0.73
	k	<b>0.43</b>	<b>948.69</b>	<b>0.84</b>	0.20

SS: Silhouette Score, CHI: Calinski-Harabasz Index, DBI: Davies-Bouldin Index, Cls: Clustering algorithm, A: Agglomerative, B: Birch, G: GMM, k:  $k$ -means

For further comparison, the clustering metrics obtained using CGAP and CPHA for different target labels are shown separately by the boxplots in Figure 2. The clustering metrics for Agglomerative, Birch, GMM, and  $k$ -means are represented by pink, blue, green, and purple box plots respectively. The y-axis for each figure in a row is common, where each tick represents one of the 11 target labels. The x-axis represents one of the three clustering metrics. The boxplot edges correspond to the 25th and 75th percentiles, the whiskers show the extreme values, and the dots highlights the outliers. Figure 2 confirms the same results as Table 1, where Agglomerative and Birch show similar or slightly lower clustering metrics for all the target labels, as compared to  $k$ -means. Meanwhile, the GMM performs worst in all cases.

CGAP and CPHA methods can also be compared based on the clustering metrics in Figure 2 and Table 1. A common trend is observed where CPHA yields higher CHI and lower DBI than CGAP, suggesting better cluster compactness and separation with CPHA. On the contrary, SS is smaller using CPHA than CGAP, suggesting some loss in overall cluster distinctness. Nevertheless, in all the cases,  $k$ -means outperforms the other algorithms.

These results suggest that  $k$ -means produces more distinct and consistent clusters. Although Agglomerative and Birch produce similar results, the rest of the evaluation focuses only on  $k$ -means for clarity and space constraints. Full results for all algorithms are available on our GitHub project page: <https://github.com/AlexandreLamb/Clustering-for-Explainability>.

Table 2 compares the impact of varying the number of extracted concepts on the clustering metrics. For CGAP, increasing  $N_{concepts}$  from 3 to 9 resulted in a decreased SS and CHI, indicating less distinct and more overlapping clusters. Conversely, for CPHA, it

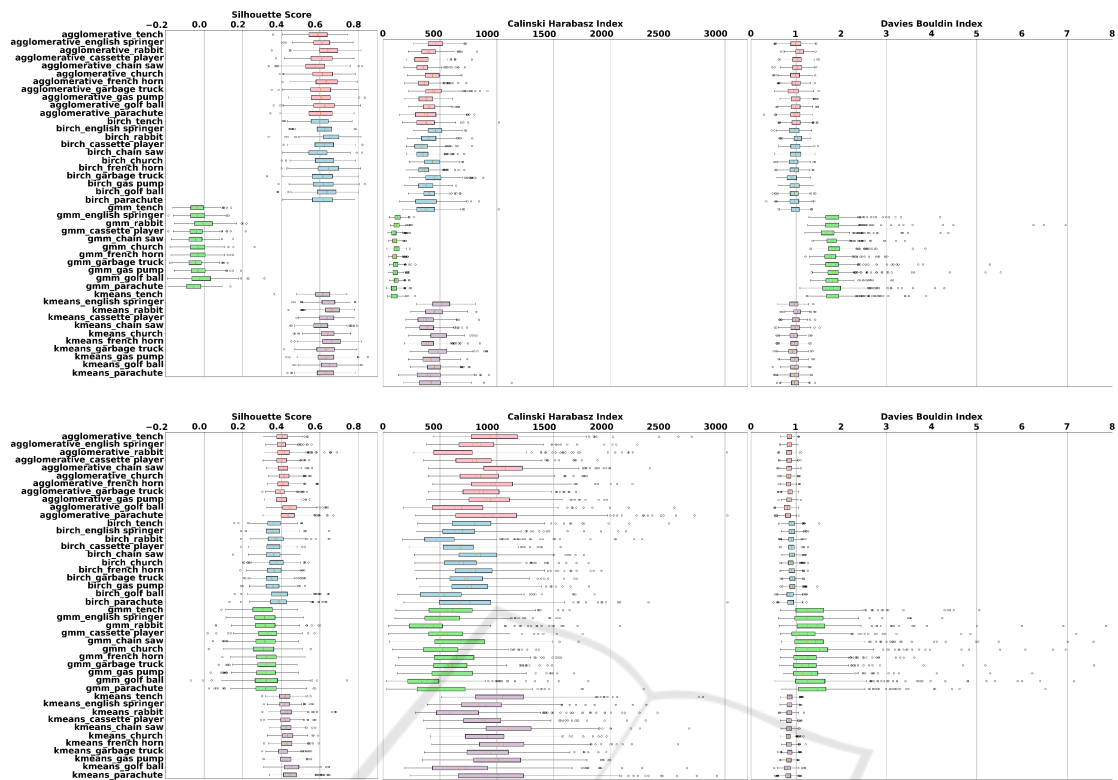


Figure 2: Clustering metrics for Agglomerative (pink), Birch (blue), GMM (green) and *k*-means (purple) on different labels. The 4 clustering algorithms are compared using CGAP (top row) and CPHA (bottom row).

led to increased SS and CHI, suggesting more distinct clusters. On the other hand, DBI does not show any specific pattern. It varies around the same range of values, implying limited usefulness in our study. Based on these observations, to achieve high-quality clusters, a smaller number of clusters is desirable for CGAP, while a large number is preferable for CPHA. In this study, we arbitrarily chose  $N_{concept} = 5$ .

Table 2: *k*-means CGAP with PCA (mean overall label).

Method	$N_{concept}$	SS	CHI	DBI
CGAP	3	0.75	966.00	0.72
	5	0.64	441.05	0.97
	7	0.57	288.34	1.10
	9	0.52	218.93	1.17
CPHA	3	0.43	900.18	0.85
	5	0.43	948.69	0.84
	7	0.44	979.13	0.83
	9	0.46	1018.18	0.81

SS: Silhouette Score, CHI: Calinski-Harabasz Index, DBI: Davies-Bouldin Index

### 4.3 Concept Visualisation and Interpretation

In this section, a visual representation of the extracted concepts is presented using the visualisation method proposed in Section 3.4. For clear visualisation, the input colour images are transformed to grayscale and the normalised activation values from concepts are used to weight the original image and are projected using the "HOT" colourmap of openCV (Itseez, 2015). As a result, the concepts are projected with a colour scale in shades of blue, where bright blue represents higher activation. For each concept  $C_l$ , the number of activations ( $A_N$ ) within  $C_l$  and the concept importance  $I_l$  are also presented. The concepts are sorted by decreasing order of  $I_l$ .

#### 4.3.1 Concept Visualisation Based on General Activation Pattern

Figure 3 visualizes 5 concepts extracted using CGAP for an image labelled "Garbage Truck". These concepts highlight key general activation patterns used by the model to predict the input image as a garbage truck. The first three concepts ( $C_1$ ,  $C_3$  and  $C_2$ ) high-

light the different garbage truck regions, ex. chassis, driver’s compartment and garbage container, with importances of 46.2, 33.296, and 16.439, respectively.

The remaining activations are clustered into concepts  $C_0$  and  $C_4$ .  $C_0$  contains relatively larger patterns, including the garbage truck and its surroundings, with an importance of 13.948. Recall that  $I_0$  represents the average importance of all activations within  $C_0$ . However, such large activation patterns can be decomposed into smaller clusters if the importance of the small cluster is of interest, using the sub-clustering proposed in Section 3.2.1. Figure 4 shows the sub-concepts obtained by decomposing  $C_0$ . The sub-clusters reveal that the activations representing the garbage truck ( $C_{01}$ ) have a higher importance of 12.709, compared to 0.153 and 1.086 for the surroundings ( $C_{00}$  and  $C_{02}$ ). This sub-clustering confirms that the model prioritizes relevant concepts for predicting the garbage truck.



Figure 3: Concept visualisation using the CGAP on an image labelled "Garbage Truck" with  $N_{concept} = 5$ .



Figure 4: Sub-clusters of concept  $C_0$  in Figure 3.

The low importance of the surrounding areas, represented by concepts  $C_4$ ,  $C_{00}$ , and  $C_{02}$ , is noteworthy and may be attributed to potential similar backgrounds in the training data, which the model associated as a relevant concept (Fel et al., 2023). The impact of these concepts on model predictions varies by application, but the importance metric helps estimate their influence. Figure 5 provides additional examples of such concepts. For the church, concept  $C_0$  initially seems to assign high importance (25.329) to the upper part of the cross. But, decomposing  $C_0$  reveals sub-concepts ( $C_{02}$  and  $C_{01}$ ) where the activations highlighting the cross have the importance of 15.698 and 9.97, while the background ( $C_{00}$ ) has negative importance of -0.339. As stated earlier, a negative influence means that it makes the model’s prediction less certain. Similarly, in the parachute example, the sub-concept ( $C_{00}$ ), including the parachute and a statue, has an importance of 40.89, whereas the sub-concepts ( $C_{01}$  and  $C_{02}$ ) including only the statue

have negative importance. Further decomposition of  $C_{00}$  could separate the parachute’s importance, though this might introduce redundant sub-concepts.

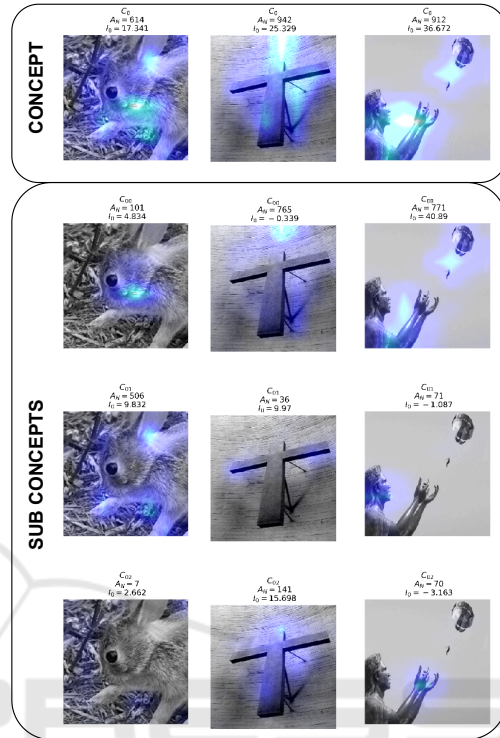


Figure 5: Concepts decomposition into three sub-concepts for different classes (rabbit, church and parachute).

### 4.3.2 Comparing CGAP and CPHA

Figure 6 compares chainsaw image concept extraction in CGAP (top) and CPHA (bottom). The most evident observation is CPHA’s capacity to extract non-redundant concepts. For CGAP, the essential concept is  $C_2$  with  $I_2 = 38.66$  highlighting the wood log and the chainsaw, which aligns well with this class. The  $C_0$  with  $I_0 = 10.022$  also highlight the same area but in a more disparate way. The other three concepts ( $C_4$ ,  $C_1$  and  $C_3$ ) redundantly focus on the chain saw engine with a cumulative importance of 65.155. In contrast, the CPAH identifies the chainsaw engine as the most important concept,  $C_1$ , with  $I_1 = 68.757$ , similar to the combined importance of the three CGAP concepts. CPHA also isolates the wood log into separate concepts ( $C_3$  and  $C_0$ ) with importances of 10.32 and 1.002, and highlights the chain and log interaction ( $C_2$  and  $C_4$ ) with a cumulative importance of 33.735.

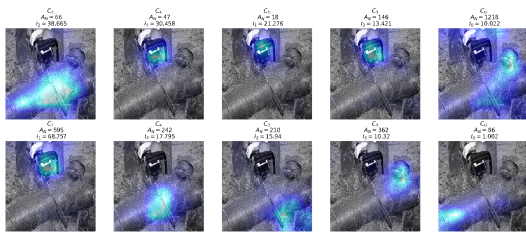


Figure 6: Concept visualisation using the CGAP (top row) and CPHA (bottom row) on an image labelled "Chain saw".

## 5 CONCLUSION

Analyzing and visualizing concepts is key to understanding model predictions. By clustering activations with similar patterns, we gain insights into the model's learned knowledge. We use two methods for concept extraction: CGAP, which focuses on general activation patterns, and CPHA, which targets high activation areas. Decomposing concepts into sub-concepts helps avoid mixing conflicting elements and compensates for clustering imperfections.

Our approach is limited by its focus on individual images, neglecting relationships between activations across images. Future work could explore clustering within the same class. While our method highlights relevant image parts for classification, incorrect classifications still require human interpretation.

## ACKNOWLEDGEMENTS

We appreciate the ECE for funding the Lambda Quad Max Deep Learning server, which is employed to obtain the results in the present work.

## REFERENCES

Atakishiyev, S., Salameh, M., Yao, H., and Goebel, R. (2024). Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions.

Brock, A., De, S., and Smith, S. L. (2021a). Characterizing signal propagation to close the performance gap in unnormalized ResNets.

Brock, A., De, S., Smith, S. L., and Simonyan, K. (2021b). High-Performance Large-Scale Image Recognition Without Normalization.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Ding, C. and He, X. (2004). K-means clustering via principal component analysis. In *Twenty-First International Conference on Machine Learning - ICML '04*, page 29, Banff, Alberta, Canada. ACM Press.

Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., and Serre, T. (2023). CRAFT: Concept Recursive Activation FacTORIZATION for Explainability.

Ghalebikesabi, S., Ter-Minassian, L., Diaz-Ordaz, K., and Holmes, C. (2021). On Locality of Local Explanation Models.

Ghorbani, A., Wexler, J., Zou, J., and Kim, B. (2019). Towards Automatic Concept-based Explanations.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA. IEEE.

Itseez (2015). Open source computer vision library. <https://github.com/itseez/opencv>.

Kim, S. S. Y., Meister, N., Ramaswamy, V. V., Fong, R., and Russakovsky, O. (2022). HIVE: Evaluating the Human Interpretability of Visual Explanations.

Lambert, A., Soni, A., Soukane, A., Cherif, A. R., and Rabat, A. (2024). Artificial intelligence modelling human mental fatigue: A comprehensive survey. *Neuro-computing*, 567:126999.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096.

Petsiuk, V., Das, A., and Saenko, K. (2018). RISE: Randomized Input Sampling for Explanation of Black-box Models.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. (2016). Grad-CAM: Why did you say that? In *NIPS*. arXiv.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.

Sivanandan, R. and Jayakumari, J. (2020). An Improved Ultrasound Tumor Segmentation Using CNN Activation Map Clustering and Active Contours. In *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pages 263–268, Greater Noida, India. IEEE.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. (2017). SmoothGrad: Removing noise by adding noise.

Wickramanayake, S., Hsu, W., and Lee, M. L. (2021). Comprehensible Convolutional Neural Networks via Guided Concept Learning.

Zhang, R., Madumal, P., Miller, T., Ehinger, K. A., and Rubinstein, B. I. P. (2021). Invertible Concept-based Explanations for CNN Models with Non-negative Concept Activation Vectors.

Zhang, Y., Weng, Y., and Lund, J. (2022). Applications of Explainable Artificial Intelligence in Diagnosis and Surgery. *Diagnostics*, 12(2):237.