THESIS

HUMAN TEACHABLE CONCEPT HIGHLIGHTER FOR POST-HOC VISUAL EXPLANATIONS

Submitted by

Erfan Mirhaji

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2026

Master's Committee:

Advisor: Sarath Sreedharan
Co-Advisor: Nathaniel Blanchard

Jill Zarestky

ABSTRACT

HUMAN TEACHABLE CONCEPT HIGHLIGHTER FOR POST-HOC VISUAL EXPLANATIONS

In recent years, deep learning (DL) models have surpassed human experts in a variety of tasks. However, when it comes to educational contexts, human experts possess something these high-performing models currently lack—the ability to provide clear and approachable explanations tailored to human learners. Efforts to develop explainable methods for these models have typically been designed for DL-experts rather than learners. In this study, we propose a novel approach that combines high-performance models with the teachability of human expert explanations. Specifically, we focus on generating post-hoc, human-understandable explanations for key expert-defined concepts on a novel task: training citizen scientists to identify pollinators from images. Our method, HuTCH, transforms the representational space and highlights relevant segments of an image for learners based on essential concepts — for example, automatically highlighting hair in an image as a key concept for bee identification. We compare HuTCH's performance by comparing against traditional saliency maps and expert annotations, and show that HuTCH concepts better align with expert annotations. The proposed framework bridges the gap between models' accuracy and human-teachable features, contributing to the advancement of explainable AI for use in pedagogy.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Sarath Sreedharan. I have learned an immense amount during my time in his lab, the HAPI lab. Through working on challenging problems in the AI space under his supervision, I gained deep knowledge of deep learning, explainable AI, and computer vision. Beyond academic knowledge, I also learned invaluable lessons about life and kindness. I always looked forward to our in-depth conversations about academia and life.

I am deeply grateful to my co-advisor, Dr. Nathaniel Blanchard, for teaching me so much about research and computer vision. I would also like to thank the members of the Native Bee Watch project: Dr. Jill Zarestky, for agreeing to serve on my committee and for her guidance on the project; Lisa Mason, for her invaluable expertise on bee characteristics; and Dr. Nikhil Krishnaswamy.

To all the wonderful members of the HAPI Lab at CSU: Malek Mechergui, Turguy Caglar, Kelsey Sikes, Septia Rani, Phil Hopkins, Dennis Kim, Brittany Cates, Trisha Ghali, Kazim Abrar Mahi, Roya Daneshi, and Shaky Jr.; I always looked forward to our conversations in the lab and learning from your perspectives on research and life. I had an amazing time working alongside you.

I would like to thank the incredible people I met at Colorado State University: Omar Soliman, Kedrick Kinsella, Lino Barrios, Alexander (AJ) Leichner, and Videep Venkatesha. You enriched my experience of living in Colorado and brought joy to my days. I am also grateful to Jon Fisher, Hossein Razmi Bagtash, and all the other kind souls I met in Colorado and beyond.

Finally, I would like to thank my parents, without whom none of this would have been possible. Thank you from the bottom of my heart for your unlimited support and patience. I hope to make you proud.

DEDICATION

To those who forsake the comfort of certainty

for the courage of asking

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In recent years, deep learning has made rapid advancements, but these advancements have been accompanied by concern and kvetching about deep learning models lack of explainability. Of course, there have been various efforts to incorporate explanations into deep learning models — but few have focused on creating explanations targeted at learners hoping to capitalize on the success of deep learning [1]. Here, we introduce a novel method for generating explanations from visual data to do exactly that — educate learners.

Traditionally, the use of CNNs for teaching novice learners has been challenging, to say the least, because of the difficulty identifying and interpreting the features that any given CNN uses. The major gap this work seeks to fill is that post-hoc explanation methods for CNNs are typically targeted at understanding the models themselves—even if a learner used them and understood the internal workings of the deep learning model, the features and concepts the CNN used when making decisions would not necessarily align with what a human experts considered relevant when they make decisions, nor what those experts would consider relevant for novices to learn.

Nonetheless, the potential to capitalize on CNNs to generate human-expert level information for learners is enticing. We were motivated to make use of the high accuracy of CNNs with the teachability of task-specific concepts via human expert feedback. In this work, we introduce a novel method called HuTCH which generates explanations that do not assume familiarity of the user with explained concepts [1]. HuTCH generates explana-

[1]The work in this thesis is based on: E. Mirhaji, N. Krishnaswamy, J. Zarestky, L. Mason, S. Sreedharan, and N. Blanchard, "HuTCH: Human Teachable Concept Highlighter for Post-hoc Visual Explanations," in *Artificial Intelligence in Education: 26th International Conference, AIED 2025*, Palermo, Italy, 2025, pp.

tions that use human-understandable concepts — refined with input from field experts — while also highlighting the region where the concept is present. The visual explanations that HuTCH produces, which could be considered illustrative explanations, are suitable for teaching applications — for this work, we use HuTCH to generate information for citizen scientists learning to identify insects. However, our method is both flexible and scalable, making it suitable for a wide range of visual learning tasks.

446–453.

Chapter 2

RELATED WORK

Post-hoc explanations have been popular in generating explanations for vision models. Gradient based explanations are one type of post-hoc explanations [2]. These methods operate by calculating the gradient of the class score with respect to image that is the input to the model. Saliency maps are one example of such methods [3]. Other methods like LIME create perturbed versions of the input image and look for change in the model's predictions [4]. Many methods besides LIME have been established to interpret complex models; however, it is often unclear when to choose one method over the other. To address this, Shapley Additive Explanations (SHAP) framework has been suggested [5]. It assigns each feature a value of importance for a particular prediction. Despite the previous efforts, LIME and SHAP, while having different definitions of attribution, can generate misleading explanations on the true reason for a prediction by a model [6], [7], [8], [9]. The assumption in all of the mentioned methods is the familiarity of the user with vision models and the features they use for classification. Many of the explanation methods mentioned above are most likely not understandable by non-experts, and even worse, not teachable to learners with little to no background in AI or understanding of features attributions. We, on the other hand, make an effort to generate post-hoc teachable explanations suited for novice learners who don't necessarily have an understanding of vision models.

Concept based explanations are another type of post-hoc explanations. In concept based explanation, explanations go beyond features of each image and identify higher level human-understandable concepts that are true for the entire dataset [10]. Concept Activation Vectors (CAVs) are another method used in concept based explanations to provide clarification about

a neural net's internal state in terms of human-friendly concepts [11]. Because of the human-explainability focus of these works, they are easier to be understood by humans. TCAV and its explainability implications has been widely used in medical applications [1]. In [12] TCAV was used to demonstrate the clinically known biomarkers that were related to cardiac disease in their model. In another study, regression concept vectors were added to TCAV. This allows for modeling concepts that vary continuously, instead of being present or absent [13]. They have shown how this approach can clarify why a network differentiates between cancerous and healthy regions. Some other methods suggest configuring the architecture of the model to achieve Post-hoc Concept Bottleneck models (PCBMs) [14]. They turn any model into a PCBM while retaining model performance and adding interpretability benefits. These explanability methods also assume the familiarity of the user with the defined concepts. If a layman is presented with these explanations, they might not understand them, as a familiarity with said concepts is implicitly expected by concept based explanations. In addition, even if the concepts are understandable by the user, they might not be able to identify the said concept in the image, leading to confusion, specially in teaching applications.

Our Human Teachable Concept Highlighter (HuTCH) method on the contrary, produces explanations based on human understandable concepts with the aid of a field expert, and further highlights the region containing the concept. We separate the decision making process of the model and the explanations that are generated by our model, which are highly teachable and interpretable to learners. Our approach is highly flexible and scalable, suitable for use in many different visual learning tasks.

Chapter 3

METHODOLOGY

In this paper, we highlight a region of an input image containing a teachable concept to learners based on information from a human expert using HuTCH framework (§ 3.1). The human expert provides both appropriate teachable concepts for each class of images and examples of said concepts (§ 3.2). Using the provided information, Concept Activation Vectors (CAVs) are calculated and then used by our method (§ 3.3). Our method first classifies the image using the specialized CNN and selects the appropriate concept based on the image class. Then, by creating sub-images from the input image, HuTCH measures the existence of the concept via the CAVs (§ 3.4). The region with most presence of the concept is highlighted and is compared with the saliency maps of the CNN as a baseline (§ 3.4 and § 3.6). We show that our method out-performs the uninformed saliency highlights, achieving higher teachability suitable for educational purposes.

3.1 HuTCH Explanation Technique

Our method attempts to bridge the notions of feature attribution explanation methods with concept explanations. As discussed, our method leverages the prediction and internal representations learned by the machine learning model as a means to generate post-hoc explanations that can be used to help teach a learner how to perform the classification on their own.

Our method starts with classification model $M : \mathbb{X} \rightarrow \mathbb{Y}$, which takes in an element from our input $\mathbb{X}$ and maps it into one from class labels $\mathbb{Y}$. Similarly, we assume access to a method

Procedure 1: An algorithm sketch describing our method to identify the relevant rule and concept explanations for a given model prediction.

Input : $M, X, \mathcal{R}, \mathbb{C}, \tau, K$
Output: $\mathcal{R}^{'}, \{(c_i,)\{\mathcal{F}_i^1, .., \mathcal{F}_i^K\}\}$
$\mathcal{Y} \leftarrow M(X)$;
$max\_prob \leftarrow 0$;
for $\tilde{\mathcal{R}}^{\mathcal{Y}} \in \mathcal{R}(\mathcal{Y})$ **do**
  $P(\tilde{\mathcal{R}}^{\mathcal{Y}}|X) \leftarrow \mathsf{get\_prob}(X, \tilde{\mathcal{R}}^{\mathcal{Y}}, \mathbb{C}(\tilde{\mathcal{R}}^{\mathcal{Y}}))$;
  if $P(\tilde{\mathcal{R}}^{\mathcal{Y}}) > max\_prob$ **then**
    $max\_prob \leftarrow P(\tilde{\mathcal{R}}^{\mathcal{Y}})$;
    $R^* \leftarrow \tilde{\mathcal{R}}^{\mathcal{Y}}$
  end
end
$top\_k\_set \leftarrow \{\}$;
if $max\_prob > \tau$ **then**
  for $\mathcal{C} \in \mathbb{C}(R^*)$ **do**
    for $\mathcal{F} \in \mathsf{feature\_extractor}(X)$ **do**
      $Q \leftarrow Priority\_Queue()$ $Q.push(\mathcal{F}, P(R^*|\mathcal{F}))$
    end
    $top\_k\_features\_for\_concept \leftarrow Q[:K]$
     $top\_k\_set = top\_k\_set \cup \{(\mathcal{C}, top\_k\_features\_for\_concept)\}$
  end
  return R^*, top_k_set
end
return $\emptyset, \emptyset$

$\mathsf{feature\_extractor} : \mathbb{X} \rightarrow 2^{\mathbb{F}}$, that extracts a set of constituent features from a given input $\mathcal{X} \in \mathbb{X}$, which can then be used in our explanations. Here, we use the notation $\mathbb{F}$ to represent the set of possible features.[1] For visual tasks, the features could include possible segments of the overall image. This form of decomposing a given input to a set of post-hoc features for explanations is a common technique used in feature attribution methods, including LIME [4].

We assume that an expert has given a set of rules $\mathcal{R}$ that can be used to teach the target learner how they can perform the classification task on their own. We denote the set of rules associated with a class $\mathcal{Y}$ as $\mathcal{R}(\mathcal{Y}) = \{\mathcal{R}_1^{\mathcal{Y}}, ..., \mathcal{R}_k^{\mathcal{Y}}\}$. We assume that all rules are defined

[1]The notation $2^{\mathbb{F}}$ corresponds to the powerset of all features provided within the set $\mathbb{F}$, thus capturing the fact that each input can be captured by a subset of feature in the set $\mathbb{F}$.

using a concept vocabulary set $\mathbb{C}$. Here, each concept $\mathcal{C} \in \mathbb{C}$ is a fact or a proposition that is either present or absent in a given input. We identify the concept using its label and probabilistic classifier that returns the probability that the concept is present in the given input. Overloading the notation, we use the symbol, $\mathcal{C}$, to stand in for both the label and the classifier. In this paper, we generally use the concept-activation vectors or CAVs [11] as the representation of each concept. There are a few ways one could use these CAVs to generate probabilities for a concept being present. For example, one could use the magnitude projection of the activation vector or you could use the accuracy of the classifier trained as the basis for the probability.

Without loss of generality, we assume that each rule is a logical formula over the individual concepts. For example, one rule for identifying wasps may assert that the given insect is a wasp if it is yellow (corresponding to the concept is_yellow) and contains a pinched waist (has_pinched_waist). As such, the corresponding rule $\mathcal{R}_1^Y$ can be represented as: has_pinched_waist and is_yellow.

For a given input, if we have the probabilities for the individual concepts being true, the probability that the rule as a whole holds can be found by multiplying the individual probabilities. We use the notation $\mathbb{C}(\mathcal{R}_i^Y)$ to return the set of concepts used in an individual rule $\mathcal{R}_i^Y$.

Given these individual elements, our goal is to identify a rule that can explain the current classification prediction with the highest likelihood. This rule is then passed to the learner. However, the learner might not be aware of the individual concepts themselves, so in addition to finding the rule, we also go over the features in the input to identify the features that have the highest impact in determining the probability of a concept being present.

Algorithm 1 puts all of these pieces together to find the overall approach to generate these explanations. Here the algorithm takes two additional inputs, namely K and τ. K provides the number of features we need to include to illustrate a specific concept, and τ represents the minimal probability for which a rule is said to be valid. Additionally, $P(\tilde{\mathcal{R}}^{\mathcal{Y}}|X)$ represents

the probability a rule is true, given an input X, Q is a priority queue that is used to track the features for which the concept classifiers returns the highest probability. At the end of iterating over the individual features, $Q[:K]$ returns the top-k features from the priority queue. The probability parameter τ provides us with a certain level of robustness against exposing the learner to incorrect classifications made by the original machine learning model.

3.2 Concepts, Datasets and Model

A visual identification task pertinent to teaching citizen scientists is the classification of bees vs wasps ($\mathbb{Y} = \{\text{bee, wasp}\}$). Entomologists look for specific characteristics when encountering an insect that is either a bee or a wasp; for example, the existence of hair on the body, or the narrowness of the abdomen and thorax connection, commonly known as a "pinch-waist". These two are some of the most important concepts differentiating bees and wasps; many wasp species exhibit this pinch-waist feature, and many bee species have hair on their body. Naturally, these concepts are suitable to be taught to citizen scientists in order to aid them in this specific visual identification task. As a result we tune our explainability method on these two features, so $\mathbb{C} = \{\text{body-hair, pinch-waist}\}$. As we only focused on one concept per taxon, we conclude $\mathcal{R}(\text{bee}) = \{\text{body-hair}\}$ and $\mathcal{R}(\text{wasp}) = \{\text{pinch-waist}\}$.

The data used in this study is gathered from the iNaturalist website [15].

We specifically chose bee species that display hair on their body, and wasp species that have the pinch-waist feature. In total we selected 23 bee species and 17 wasp species, and 226,892 images of bees and 229,921 images of wasps were collected. We split the data into train/test splits, using 85% for training the CNN and 15% for testing, defining the concept dataset and comparing methods.

In order to highlight regions containing a certain concept using our HuTCH method, we first define a concept dataset with 200 bee images and 200 wasp images from our testing dataset. A human expert selected one region in each image that contains the corresponding concept, and one region where the concept is absent. From this, we had two datasets: one

for each concept, containing 200 positive and 200 negative examples. Figure 3.1 demonstrates an example of expert concept selection. Next, by applying six scaling values and four ninety degree rotations on both the original and flipped concept examples, we augmented our dataset, culminating in 9,600 positive and 9,600 negative samples.

We used a total of 120 testing images, 60 bee images and 60 wasp images from our testing dataset for our comparisons ($\|X\| = \|\text{testing-images}\| = 120$). First, the original images are resized to 224 by 224 and then segmented using the Mask R-CNN ResNet-50 object instance segmentation method [16] with zero threshold to find the largest segment. The results are then given to the expert, the saliency highlighter, and our two HuTCH highlighter methods, each outputting their highlights.

Figure 3.1: Examples of the concept dataset. The left-most column displays a subset of the total 400 images given to the expert, showing a bee at the top and a wasp at the bottom image. The middle column shows the regions that the expert has identified as positive examples, while the right-most column features regions that do not contain the concept.

3.3 Concept Activation Vectors

CNNs transform images to higher representational spaces and extract many complex features. In the case of ResNet-152, we use the output of the second convolutional layer of the third bottleneck layer of the fourth residual stage, which contains 512 features each with a spatial dimension of 7×7. We call this layer our layer of interest, which is the second last convolution layer of the entire architecture.

We then pass each of our concept datasets to the model and train a linear classifier on the flattened activations of our layer of interest. The vector containing weights of this linear classifier is the Concept Activation Vector (CAV). Later on, using the dot product of the flattened activations of a new image and the CAV of a concept, we can calculate the alignment of that image with that concept.

3.4 Human Expert and CNN Highlights

Using the selected 120 test images, we ask the expert, the CNN model and our HuTCH method to highlight what each think is the most important region. The human expert highlights the relevant concept region in each image; in the case of a bee the hair region, and in the case of a wasp the pinch-waist region. Only one continuous region is highlighted by the expert.

Saliency maps reflect the degree of importance of a pixel to the black-box model. The CNN model, using the normalized gradients with a threshold of 0.05, calculates the saliency map of each image. The acquired saliency map of each image is then further partitioned into rectangles. Each map is partitioned into rectangles of widths and heights of 56, 74 and 112, creating 100 partitions. Then, the partitions are saved separately to create new images referred to as sub-images. The average gradient value of all sub-images are then measured and sorted. The sub-images corresponding to the top K highest average gradient values are overlapped and combined together and that region is the highlighted region of the saliency

map.

3.5 HuTCH Highlights

Our method, on the other hand, highlights regions in which a certain concept is most present. In our HuTCH framework, each image is first passed to the CNN model to make a prediction, and whether it is a bee or a wasp, the concept to look for is set to hair or pinch-waist respectively; this is done to combine the accuracy of CNNs with the teachability of expert-defined concepts. We then further segment the image with 0.05 threshold to filter out the background from the object of interest, in this case the insect. Next, using two different methods, HuTCH further partitions the image; once using rectangles, called HuTCH Rectangle, and another time using all the segments generated by the Mask R-CNN segmentation method [16], called HuTCH Segmented.

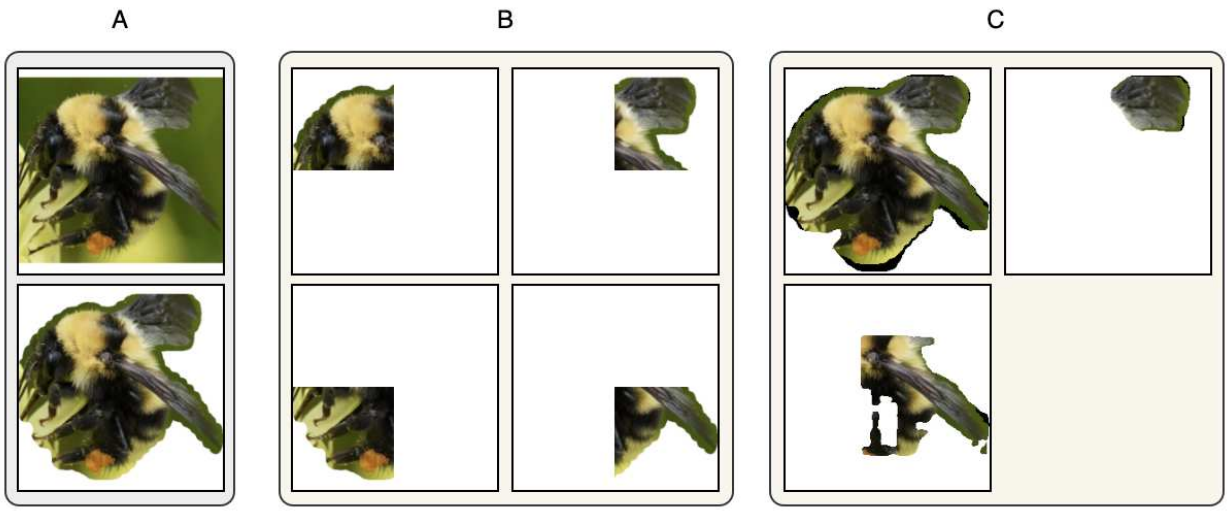

Figure 3.2: An example of sub-images created by HuTCH. **A** shows the original input image at the top, and the largest segment with 0.05 threshold used by both HuTCH methods at the bottom. **B** shows the 112 by 112 rectangles created by HuTCH Rectangle, and **C** demonstrates the segments created by the R-CNN model used by HuTCH Segmented. Note that the R-CNN model only recognized three masks, so we end up with three segments extracted from the image. The top K sub-images corresponding to the top K CAV dot products are then overlapped and form the highlighted region by HuTCH.

Depending on the image, the R-CNN model can generate anywhere from 2 to 21 segments.

In the HuTCH Rectangle, similar to the saliency partitioning, the image is partitioned into all possible rectangles with sides of 56, 74 and 112 pixels. Again, we refer to each segment and each partitions as a sub-image. The feature extractor is the method we use to create the sub-images, either rectangles or R-CNN segments ($sub - image \in \mathsf{feature\_extractor}(image)$). These steps dissect every single image into two datasets, one dataset of sub-images of HuTCH

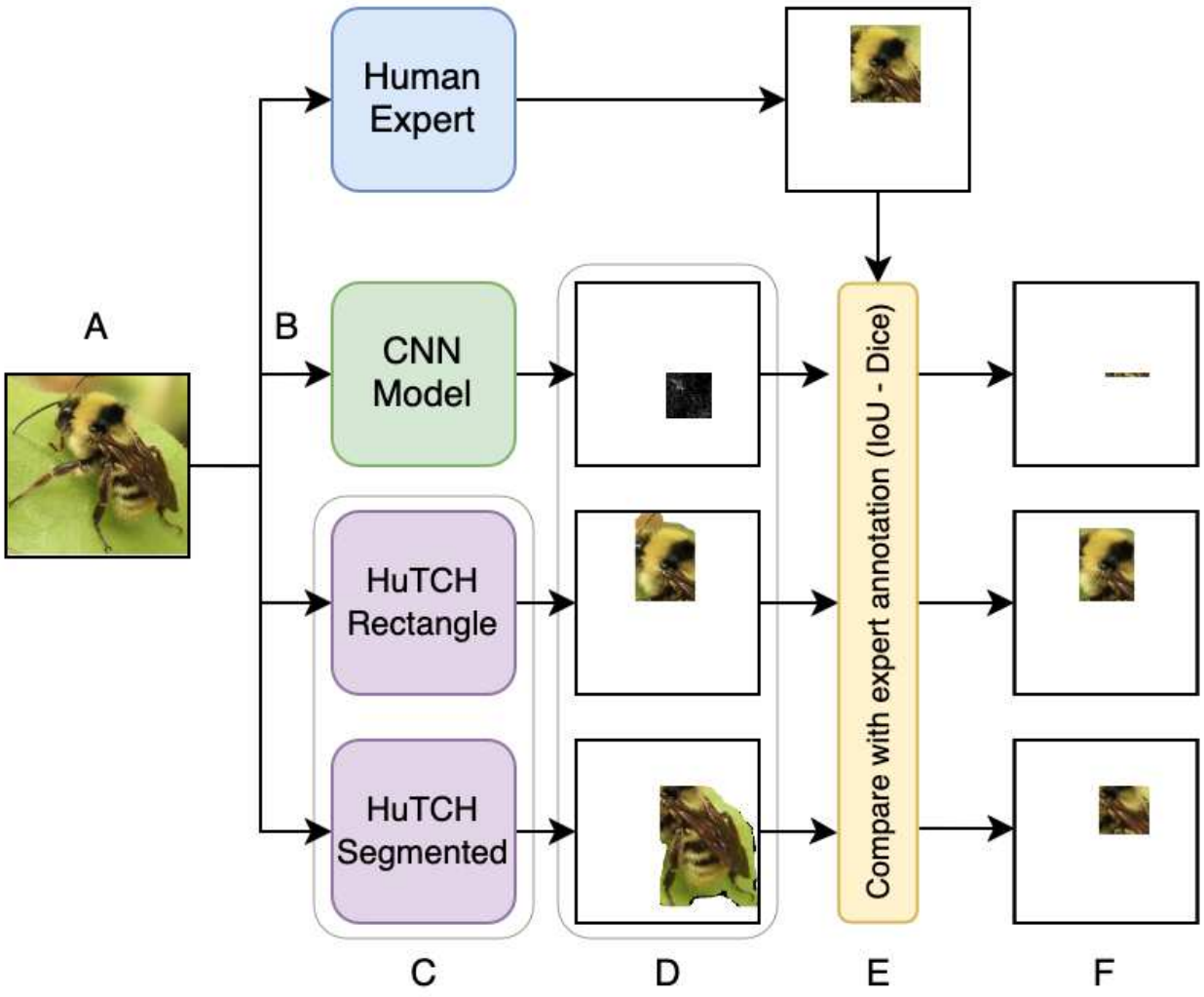

Figure 3.3: The workflow of highlighting images by each method. **A** Each input image is filtered to the biggest object segment with threshold of zero. **B** The CNN model calculates the saliency map, and sorts the rectangles with highest average gradient. **C** The two HuTCH Rectangle and HuTCH Segmented methods partition the image into sub-images via rectangles and masks respectively, and calculate the sub-images with the highest dot CAV scores. **D** The combined region of top K sub-images is highlighted by each of the three methods. **E** The highlights are compared to the expert's highlighted region via IoU and Dice metrics. **F** The overlapping region is shown.

Rectangle, one dataset of sub-images of HuTCH Segmented. Figure 3.2 shows an example of the sub-images created by each HuTCH method. Finally, each sub-image of each HuTCH dataset is passed through the model to acquire the activation from the layer of interest. The dot products of the flattened activations and the CAV of the respective concept are then measured and sorted [11]. The sub-images corresponding to the K highest dot products are merged together, making a bigger sub-image, and that region is the highlighted region of each HuTCH method. In the case of top 1 region, only the sub-image with the highest dot product remains and it is not merged with any other partition.

3.6 Comparisons

In order to compare the highlighted region from each method (Saliency Rectangle, HuTCH Rectangle and HuTCH Segmented) with the highlighted region by the expert, we used two metrics; Intersection over Union (IoU) and Dice coefficient. The highlighted region by the saliency map, the HuTCH Rectangle and HuTCH Segmented methods are compared to the region that expert has marked as the region of concept which is relevant to correctly identifying the insect. The overall workflow can be seen in figure 3.3. The results of comparisons can be viewed in Tables 4.2 and 4.3 and figure 4.1.

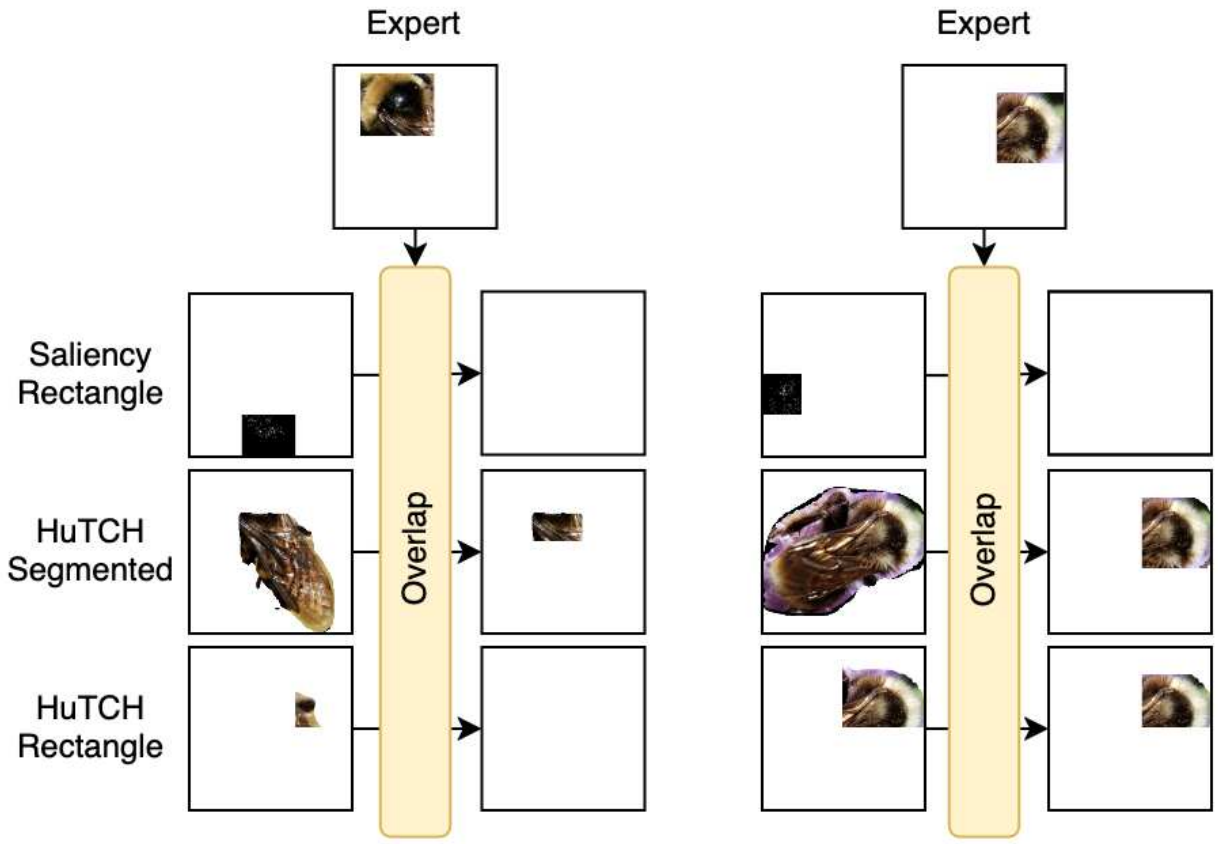

Figure 3.4: An example of comparing highlights of the three methods with expert highlight. Each of the three Saliency Rectangle, HuTCH Segmented and HuTCH Rectangle methods highlights the most important region. Then, by overlapping each region with the expert region, we can quantify the alignment. This figure only demonstrates the overlap; IoU and Dice are calculated separately.

The concept that the expert highlights is something that is understandable by humans, and is used to teach about different taxa in entomology. After doing this comparison, we can conclude which method is closest to expert methodology when it comes to teachability. We show that both our methods are superior to the saliency maps, which means our methods are closer to human teachability. By combining the teachability of our methods and the high accuracy of CNNs, we can teach learners with high certainty and high explainability.

Chapter 4

EXPERIMENTS & RESULTS

In this section, we first choose the CNN and evaluate the performance of the chosen model and compare the performance of traditional saliency maps and our HuTCH methods with our expert annotations.

4.1 CNN Performance

For the CNN model, we chose the ResNet-152 architecture [17], and the model was re-trained as a whole, updating all the weights (M = ResNet-152). We calculated the mean and standard deviation values of our training dataset and adjusted the input images accordingly. The trained ResNet-152 CNN model was tested on 34,022 bee and 34,481 wasp images and it achieved a combined F_1 score of 96 percent. Table 4.1 presents additional details on the performance of the model.

Table 4.1: Classification performance of the ResNet-152 model on test dataset.

Taxon	Precision	Recall	F1-score	Support
Bee	0.98	0.95	0.96	34022
Wasp	0.95	0.98	0.97	34481
Macro avg	0.96	0.96	0.96	68503
Weighted avg	0.96	0.96	0.96	68503

4.2 HuTCH vs Saliency Performance

As discussed in section 3.6, we used two metrics of Intersection over Union (IoU) and Dice coefficient on 120 test images to compare the highlights of each method with expert annotations. Tables 4.2 and 4.3 show the average and standard deviations of IoU and Dice values for each method on the aggregated top K regions respectively. Figures 4.1 and 4.2 show the average IoU and Dice coefficients along with the 95% confidence intervals.

Each of the three sub-plots in figure 4.1 represent the combined top 1, top 2, and top 3 regions. The three bars in each sub-plot stand for Saliency Rectangle (SR), HuTCH Segmented (HS), and HuTCH Rectangle (HR) from left to right respectively. As demonstrated, both HuTCH methods achieve higher IoU and Dice means compared to the saliency map. We can see that in all of the three top K plots, HuTCH Rectangle shows statistically significant improvement over Saliency highlights in both IoU and Dice metrics.

Table 4.2: The average and standard deviation of IoU scores between highlighted regions of each method and expert highlights.

Method	**Top 1**	**Top 2**	**Top 3**
Saliency Rectangle Highlight	0.155 ± 0.166	0.189 ± 0.164	0.209 ± 0.164
HuTCH Segmented Highlight	0.237 ± 0.165	0.237 ± 0.124	0.238 ± 0.112
HuTCH Rectangle Highlight	0.275 ± 0.211	0.309 ± 0.190	0.307 ± 0.161

Table 4.3: The values for Dice coefficient between highlighted regions of each method and expert highlights.

Method	**Top 1**	**Top 2**	**Top 3**
Saliency Rectangle Highlight	0.237 ± 0.288	0.288 ± 0.220	0.317 ± 0.216
HuTCH Segmented Highlight	0.356 ± 0.212	0.368 ± 0.159	0.372 ± 0.140
HuTCH Rectangle Highlight	0.388 ± 0.263	0.439 ± 0.230	0.446 ± 0.198

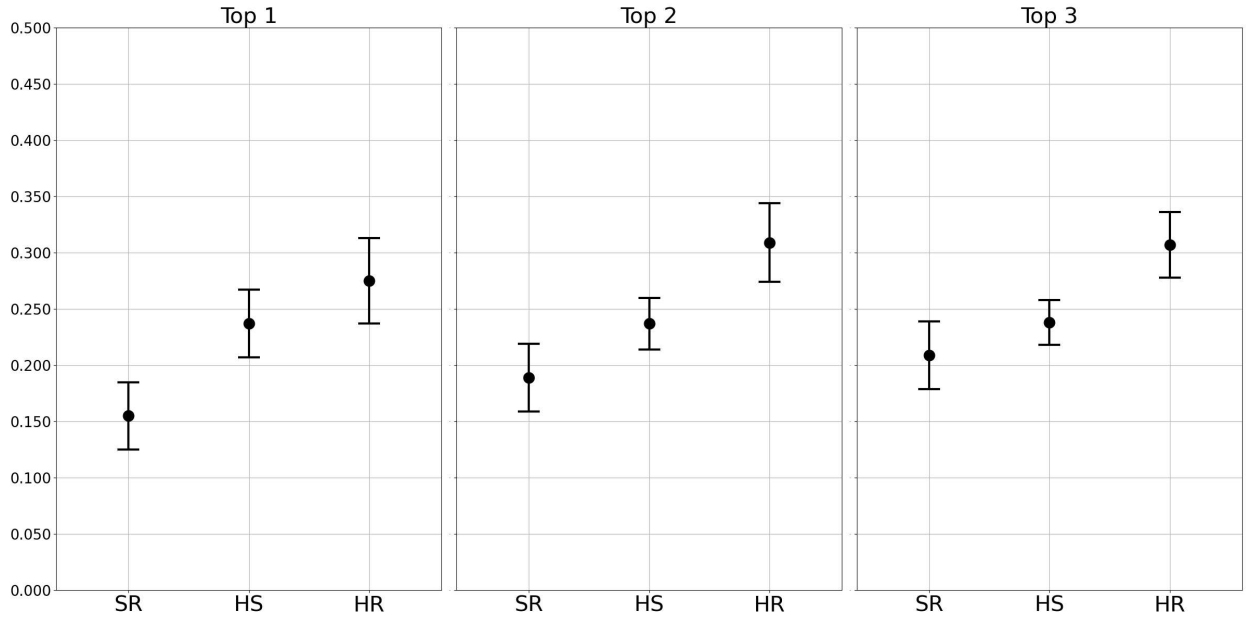

Figure 4.1: The average and 95% confidence intervals of IoU metrics. From left to right: combined top 1, top 2, and top 3 regions.

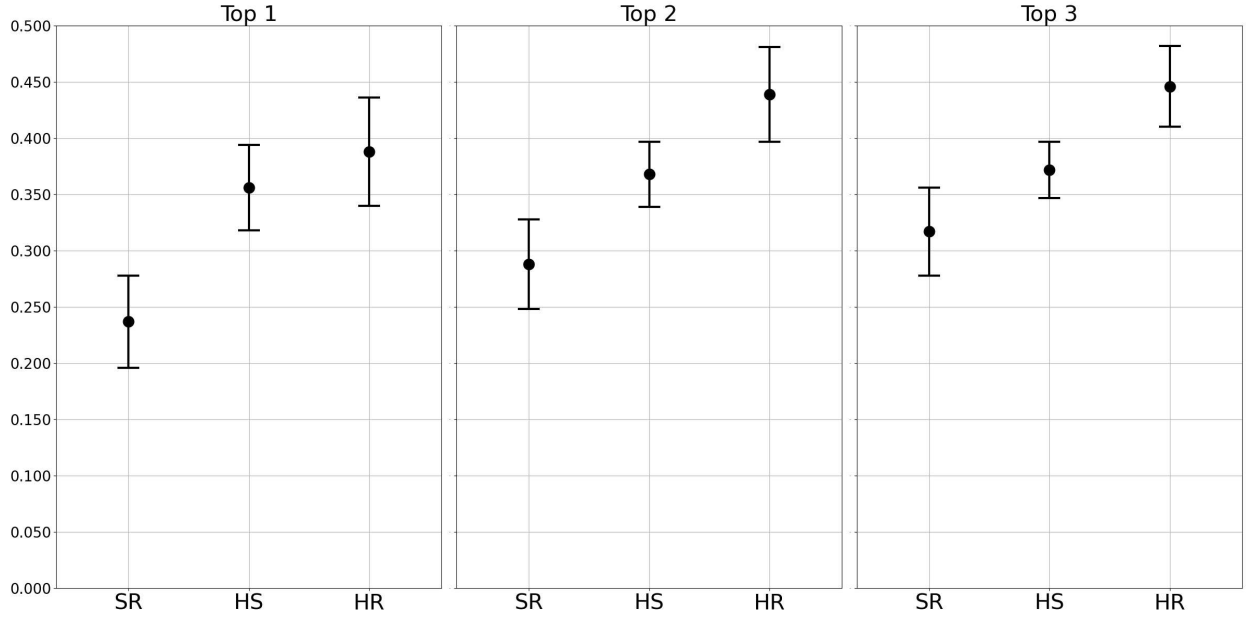

Figure 4.2: The average and 95% confidence intervals of Dice coefficients. From left to right: combined top 1, top 2, and top 3 regions.

Chapter 5

LIMITATION & FUTURE WORK

In this chapter, I discuss the limitations of my work, followed by some proposed future directions for expansion of this research.

5.1 Limitations

In this project, we used the mask R-CNN ResNet-50 model, which is trained on Microsoft's COCO dataset, to segment our images. Even though this approach was able to come up with workable masks for our segmentation needs, it had it's limitations. Sometimes unrelated background patches found their way into our sub-images which resulted in erroneous dot CAV scores. Also, more fine-tuned segmentation method that can cleanly segment bees and wasps into different segments, like abdomen, thorax, head, antennae and so on, would help us work with finer concepts that are used by human experts. The main reason for rectangle partitioning was lack of a fine-tuned segmentation technique.

Another limitation was the dataset we used. We used the iNaturalist API [15] to download thousands of images of bees and wasps. These images have been captured by amateurs and enthusiasts, sometimes lacking good focus or visibility of the insect. A more intricate and cleaner dataset would definitely improve both the classification accuracy and the relevance of CAV features that are used to train the concept classifier.

5.2 Future Work

Currently, we only used two concepts, body hair and pinch-waist, for training the CAVs and subsequently the linear classifier. We also only trained the CNN on limited bee and wasp species exhibiting these features. For future work, we can expand the concepts and the training dataset to include more features and more species showing those features. That way we can achieve species level identification. Due to limitation of the segmentation method and the dataset, it was challenging to choose good examples to test the highlighting methods. If those are improved, we can test our methods on a much bigger test dataset.

Chapter 6

CONCLUSION

In this work, the proposed HuTCH framework bridges the gap between AI models' accuracy and human-teachable features, contributing to the advancement of explainable AI in teaching. We have demonstrated that concept-based highlighting achieves more alignment with expert highlight, and thus improves the teachability of the decision-making process. We compared the highlighted regions by our HuTCH method and saliency maps, a traditional post-hoc explainability method, with expert annotations to benchmark the alignment of each method with expert highlights. Our method compares with field experts who produce approachable explanations which can be easily understood and learned by humans. Our method demonstrates how AI models' high performance can be coupled with human-understandable interpretability. Similar approaches can be used in educational settings to improve the learning experience of students when exposed to visual identification tasks.

Bibliography

[1] B. H. Van der Velden, H. J. Kuijf, K. G. Gilhuijs, and M. A. Viergever, "Explainable artificial intelligence (xai) in deep learning-based medical image analysis," *Medical image analysis*, vol. 79, p. 102 470, 2022.

[2] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," *Acm computing surveys*, vol. 51, 2018. DOI: 10.1145/3236009.

[3] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Workshop at international conference on learning representations*, 2014.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ser. KDD '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777.

[6] Y. Zhou, S. Booth, M. T. Ribeiro, and J. Shah, "Do feature attribution methods correctly attribute features?" *Proceedings of the aaai conference on artificial intelligence*, vol. 36, no. 9, pp. 9623–9633, 2022. DOI: 10.1609/aaai.v36i9.21196.

[7] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2017, pp. 6541–6549.

[8] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, pp. 3681–3688, 2019. DOI: 10.1609/aaai.v33i01.33013681.

[9] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Advances in neural information processing systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018.

[10] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, "Towards automatic concept-based explanations," in *Advances in neural information processing systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[11] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. B. Viégas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*, 2017.

[12] J. R. Clough, I. Oksuz, E. Puyol-Antón, B. Ruijsink, A. P. King, and J. A. Schnabel, "Global and local interpretability for cardiac mri classification," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2019, pp. 656–664.

[13] G. M., A. V., M.-M. S., and M. H., "Concept attribution: Explaining cnn decisions to physicians," *Computers in biology and medicine*, vol. 123, p. 103 865, 2020. DOI: https://doi.org/10.1016/j.compbiomed.2020.103865.

[14] M. Yuksekgonul, M. Wang, and J. Zou, "Post-hoc concept bottleneck models," *Arxiv preprint arxiv:2205.15480*, 2022.

[15] *Inaturalist*, Available from https://www.inaturalist.org, Accessed: January 2025.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 ieee international conference on computer vision (iccv)*, 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] C. Gütl and V. M. García-Barrios, "The application of concepts for learning and teaching," in *Proceedings of 8th international conference on interactive computer aided learning (icl 2005)*, Citeseer, 2005.

[19] E. G. Blanchard and P. Mohammed, "On cultural intelligence in llm-based chatbots: Implications for artificial intelligence in education," in *Artificial intelligence in education*, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds., Cham: Springer Nature Switzerland, 2024, pp. 439–453.

[20] Y. Yao, "Concept formation and learning: A cognitive informatics perspective," in *Proceedings of the third ieee international conference on cognitive informatics, 2004.*, 2004, pp. 42–51. DOI: 10.1109/COGINF.2004.1327458.

[21] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," 2016.

[22] Q. Ma, H. Shen, K. Koedinger, and S. T. Wu, "How to teach programming in the ai era? using llms as a teachable agent for debugging," in *Artificial intelligence in education*, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds., Cham: Springer Nature Switzerland, 2024, pp. 265–279.

[23] S. Sonkar, N. Liu, D. B. Mallick, and R. G. Baraniuk, "Marking: Visual grading with highlighting errors and annotating missing bits," in *Artificial intelligence in education*, A. M. Olney, I.-A. Chounta, Z. Liu, O. C. Santos, and I. I. Bittencourt, Eds., Cham: Springer Nature Switzerland, 2024, pp. 309–323.

[24] A. Linson, Y. Xu, A. R. English, and R. B. Fisher, "Identifying student struggle by analyzing facial movement during asynchronous video lecture viewing: Towards an automated tool to support instructors," in *Artificial intelligence in education*, M. M. Rodrigo, N. Matsuda, A. I. Cristea, and V. Dimitrova, Eds., Cham: Springer International Publishing, 2022, pp. 53–65.

[25] Y. Wang, T. Zhang, X. Guo, and Z. Shen, *Gradient based feature attribution in explainable ai: A technical review*, 2024. arXiv: 2403.10415 [cs.AI].

[26] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019. DOI: 10.3390/electronics8080832.

[27] E. Mirhaji, N. Krishnaswamy, J. Zarestky, L. Mason, S. Sreedharan, and N. Blanchard, "Hutch: Human teachable concept highlighter for post-hoc visual explanations," in *Artificial intelligence in education: 26th international conference, aied 2025, palermo, italy, july 22–26, 2025, proceedings, part v*, Palermo, Italy: Springer-Verlag, 2025, pp. 446–453. DOI: 10.1007/978-3-031-98462-4_56.