Distribution Agreement

James Song April 10, 2024

A Flask Framework for Visual Attention-Prompted Prediction

By

James Song

Liang Zhao
Advisor

Department of Computer Science

Liang Zhao
Advisor

Xiaofeng Yang
Committee Member

Chris Rodgers
Committee Member

2024

A Flask Framework for Visual Attention-Prompted Prediction

By

James Song

Liang Zhao
Advisor

An abstract of
A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciencewith Honors

Computer Science

2024

Abstract

A Flask Framework for Visual Attention-Prompted Prediction
By James Song

Deep Neural Networks (DNN) have demonstrated remarkable performances in the field of Computer Vision (CV), showing promising potentials in various areas. However, the 'black box' nature of the DNNs introduces challenges on ensuring the interpretability of such models, making it difficult to trust DNN models on fields with high stakes. In response, visual explanation-guided learning utilizes human annotated explanations in training to guide DNN model's reasoning process, making it more reasonable and trustworthy. However, this process requires a large number of explanation annotations, which takes a lot of resource to prepare, resulting in the emergence of visual attention-prompted prediction. This approach involves utilizing visual attention guidance during the application stage instead of in the learning phase, and thus eliminates the need of large amount of visual explanations and enables the direct guidance from the end user. To help facilitate this process, we propose a visual attention-prompted prediction framework that provides a user friendly application for real-time visual attention annotation and comparison between predictions with and without explanation guidance. Though the proposed framework can work with any convolutional neural networks (CNN) provided by the user, we still provide an already trained CNN model for out-of-box experience. The provided model incorporates a novel co-training process for prompted and non-prompted models, making the non-prompted model to have similar reasoning process as the prompted model. Extensive experiments on four real world datasets demonstrate the effectiveness of our provided model in situation where visual prompt is scarce. A detailed instruction of how to use our framework is also provided.

A Flask Framework for Visual Attention-Prompted Prediction

By

James Song

Liang Zhao
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciencewith Honors

Computer Science

2024

Acknowledgments

I would like to thank my thesis advisor, Dr. Liang Zhao, for his support and guidance along the way. Since I joined his lab, Dr. Zhao has greatly inspired my interests in deep learning and computer vision. I would also like to thank Dr. Xiaofeng Yang and Dr. Chris Rodgers for being on my thesis committee and providing me with valuable suggestions and feedback about my project. Thanks to my partner, Zhuoyang Zhang, who is always by my side and encouraging me, and thanks to my parents for supporting and trusting me. Those meant a lot to me. Last but not least, special thanks to Shawpaw (my cat) and Ollie (my dog) for the emotional support.

Contents

1 Introduction **1**

2 Background **4**

2.1 Attention-guided Learning 4

2.2 Attention Prompt 5

3 Approach **6**

3.1 Problem Formulation 6

3.2 Implementation 7

3.3 Overall Framework 7

3.3.1 Loading Model and Data 7

3.3.2 Annotation 8

3.3.3 Result & Evaluation 8

3.4 Model 9

3.4.1 ResNet 9

3.4.2 Visual Attention-Prompted Learning 11

4 Experiments **14**

4.1 Dataset 14

4.2 Experimental Setup 15

4.2.1 Comparison Methods 15

4.2.2 Implementation Details 15

5 Analysis **17**

5.1 Results 17

5.2 Instruction for the Visual Attention-Prompted Prediction Framework 18

6 Conclusion **24**

Bibliography **25**

List of Figures

3.1 Residual Block of ResNet 9

3.2 Structure of ResNet18 11

3.3 Structure of Visual Attention-Prompted Learning 12

5.1 Framework Page for Model Selection 20

5.2 Drop-down menu of Model Selection 20

5.3 Framework Page for Human Explanation Annotation 21

5.4 Ongoing Annotation 22

5.5 Framework Page for Prediction Result 23

5.6 Framework Ending Page 23

List of Tables

5.1 Performances of tested models on LIDC and Pancreas datasets. The best result of each metric is in bold, and the second best result is underlined [28]. 18

5.2 Performances of tested models on Gender and Scene datasets. The best result of each metric is in bold, and the second best result is underlined [28]. 19

Chapter 1

Introduction

Deep Neural Networks (DNN) have demonstrated remarkable performances in the field of Computer Vision (CV), showing promising potentials in various areas including transportation, healthcare, finance, etc. However, the 'black box' nature of the DNNs introduces challenges on ensuring the interpretability and explainability of such models, making it difficult to apply DNN models on fields with high stakes [1]. In response, the concept of Explainable AI (XAI) has emerged with the aim of interpreting and understanding the rationales behind DNN models' predictions [26]. With the development of techniques such as CAM, Grad-CAM, and integrated gradients, researchers are able to generate saliency maps, which highlight the areas on the input data that contribute most to the models' predictions [29, 24, 21], making DNN models more transparent and trustworthy.

Besides the effectiveness of XAI in interpreting DNN models' rationales, XAI also aims to enhance DNN models' predictive performances by improving their reasoning process. One of the approaches is to incorporate human expert's knowledge into the model's reasoning process and thus ensures the correctness and reasonableness of the prediction results, which is called explanation-guided learning [28]. Explanation-guided learning in CV has been explored through recent years and has been proven to be effective in improving accuracy of DNN model's predictions and reducing the disparity between the model's reasoning and the human expert's true reasoning [4, 25, 7, 8]. However, explanation-guided learning requires a large amount of training data with human-annotated attention maps, which is costly in labor, money, and time [28]. In contrast, in many practical applications, it is relatively easy for users to provide high-level guidance (visual attention prompts) for DNN model's predictions. For instance, in cancer imaging, clinicians can quickly identify areas that would potentially indicate whether the patient has cancer or not. This approach introduces the human guidance to the DNN model during the application stage instead of training stage, and thus is called attention-prompted prediction. As a result, an efficient framework that enables users to incorporate their guidance into the model's decision-making process is in need. To tackle this challenge, Zhang et al. [28] proposed the Visual Attention-Prompted Prediction and Learning framework that aims to integrate visual attention prompts into the model's decision-making process, distills knowledge of human guidance from prompted model to non-prompted model, and refine incomplete visual attention prompts.

This study focuses on implementing Zhang et al.'s previous work into a user friendly application to facilitate the real-time visual attention-prompted and non-prompted predictions. The main contributions of this paper are as follows:

- A user-friendly application based on Zhang et al.'s work, developed to enable users to incorporate human annotated visual attention prompts into their DNN model's decision-making process in real time.
- A pipeline that provides direct comparison between performances of visual attention-prompted and non-prompted models under the same setting.
- Implementation of a state-of-the-art visual attention-prompted learning model [28] as the default model for the application.

- Comprehensive analysis on experiments on real-world datasets to demonstrate the effectiveness of the framework and detailed instructions for how to use the application.

Chapter 2

Background

This section introduces previous studies related to XAI in CV, aiming to provide comprehensive background knowledge in the field.

2.1 Attention-guided Learning

The integration of human's insights into interpretable DNN models, also known as attention-guided learning, has been extensively studied in natural language processing (NLP) and CV domains. Narang et al., Hsieh et al., Raffel et al. [18, 11, 22] have trained language models using extracted rationales and labeled supervision. Meanwhile in CV, attention maps that highlight essential areas on the image, such as those generated by Grad-CAM [24], or intrinsic attention mechanism [27] have been utilized as attention guidance in model training. Combined with the prediction loss, these guidance can further improve models' performances [3, 25, 9]. Besides using attention guidance to directly improve model's performance, frameworks that help reduce the dispartiy between model's reasoning and human's true reasoning have also been proposed. Shen et al. [25] proposed a conceptual framework called HAICS that allows direct display of attention maps from convolutional neural network (CNN) and enables users to evaluate the attention maps using scribble annotations. And Gao et al. [7] developed a novel and robust model objective to handle inaccuracy, incompleteness, and inconsistency in human annotations.

2.2 Attention Prompt

Prompting originates from NLP as a method to guide language model with instructions to desired outputs [16]. Later it is adapted by Dosovitskiy et al. [5] for CV tasks. Jia et al. [12] proposed Visual Prompt Tuning (VPT), which introduces a small number of learnable parameters into the input space of Vision Transformer (ViT) model during the finetune stage. Paiss et al. [19] adopted an explainability-based approach to improve one-shot classification for images with text prompts. Li et al. [14] proposed saliency prompts generated from saliency masks to accelerate and enhance the pre-train process of instance segmentation model. Though these works all involved utilizing prompts along with original image as inputs, our work is different from them as we are using human-labeled attention map directly as the prompt.

Chapter 3

Approach

This chapter aims to provide a comprehensive demonstration of our proposed method. The chapter would start with the problem formulation to establish the concepts related to the problem and our approach. Then, the section of implementation would introduce the platforms and libraries used to develop our framework. Overall framework would provide a walk through of how to use our framework. Finally, the model section would explain the architecture of the default model in details.

3.1 Problem Formulation

In the context of visual attention-prompted learning and prediction, each sample from a dataset $\mathcal{S}$ would be in the form of $(X, P, y) \in \mathcal{S}$, where $X \in \mathbb{R}^{C \times H \times W}$ represents the original input image, with C, H, and W denotes the input image's channels, heights, and width, respectively, y is the class label corresponding to the input image, and $P \in \mathbb{R}^{H \times W}$, with identical heights and width as the input image but only one channel, denotes the visual attention prompt for the input image corresponding to its class label. The visual attention prompt P, in the form of a binary matrix, marks the areas on the input image that are particularly relevant to the prediction. Thus, the visual attention-prompted prediction process of model f can be defined as $f(X, P) \rightarrow y$.

3.2 Implementation

Based on the challenges stated earlier, we present Visual Attention-Prompted Prediction (VAPP) framework. It is a browser-based user interface that allows users to annotate their own datasets and test the annotated images with their own model. It is implemented with a lightweight back end built with Python Flask, compatible with PyTorch [20], a widely used Machine Learning (ML) and visualization libraries in Python. The front end was developed using HTML, CSS, and JavaScript, enabling dynamic user interaction including uploading models, setting tasks, and labeling images. More detailed technical settings can be found in our GitHub repository.

3.3 Overall Framework

3.3.1 Loading Model and Data

The Visual Attention-Prompted Prediction framework allows users to upload their CNN models and datasets. Currently, the framework is compatible with one of the most widely used Python libraries for building CNN, PyTorch [20]. Before starting the framework, users need to first put their models and datasets in the specified folders. Then, upon starting the framework, it would automatically scan the 'model' and 'images' files respectively for available models and images. On the initial web page, users would be asked to select the specific models for both visual prompted and non-visual prompted predictions, and set the class labels for the task they want to test. The dataset, if correctly saved in the corresponding folder, will be loaded automatically. In the case that the users do not have a specific model to test, the framework also provides a Visual Attention Prompted Learning (VAPL) [28] model with ResNet18 as the backbone as default for testing and demonstration. Similarly, if users do not specify the class label, the framework would provide the index of the class with the highest probabilities predicted by the model. Otherwise, the framework would convert the model output into corresponding class. After setting up the model and label classes, users can click the 'submit' button and move on to the next phase for annotation.

3.3.2 Annotation

If the previous stage is set up correctly, The Visual Attention-Prompted Prediction framework would continue to the next stage where it would ask users to start annotating the images from the uploaded dataset. For each image, the framework would display the original image on the left as a reference and also provide the same image as a canvas on the right. Users can annotate the image by circling what they would consider as important areas for prediction with the mouse on the image. If users are not satisfied with what they have drawn, they can use the right click to erase the strokes or click the 'Remove All' button to erase all of their drawings on the current image. Once they are done with the labeling, they can click 'Submit' button to send both the image and the user-labeled visual attention-prompt as the input to the models and check the prediction results.

3.3.3 Result & Evaluation

After users submit their customized attention-prompt, the framework would convert users' drawings into binary matrix and perform a element-wise multiplication with the preprocessed original input image, in the back end. Then, both the original image and their product would be sent into the models for prediction, respectively. Once the models generate the corresponding predicted class probabilities, the framework would convert each of them into the classes with the highest probabilities, and present them to the users. Along with the prediction results, the framework would also present the original input image and the user-labeled image, the inputs of the two models respectively, on the side for users' comparisons and evaluations.

3.4 Model

The default model provided by our framework uses ResNet18 [10] as the backbone. This section will introduce both the structures of ResNet18 and our selected model.

3.4.1 ResNet

ResNet [10] is a convolutional neural network (CNN) specifically engineered for image recognition tasks. The defining feature of ResNet is its use of residual blocks (Fig. 3.1), facilitating the training of very deep networks.

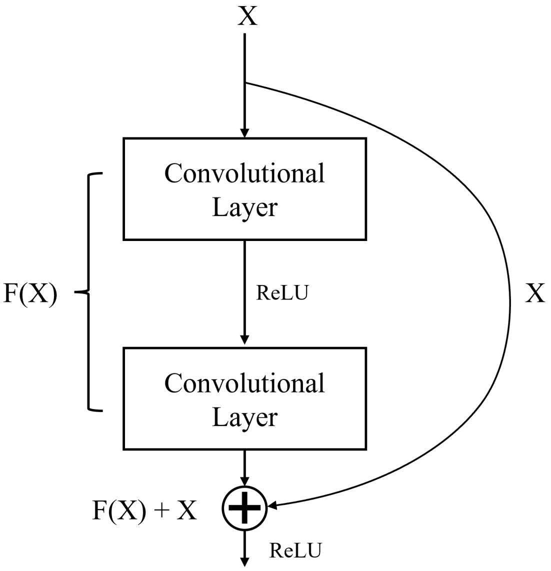


Figure 3.1: Residual Block of ResNet

As shown in Fig. 3.1, each residual block contains two 3 × 3 convolutional layers connected by a rectified linear unit (ReLU) activation. Each of the convolutional layers is followed by a batch normalization. Besides going through the two convolutional layers, the input of the residual block would also go through a shortcut connection to be added to the output of the second convolutional layer, allowing the network to learn residual mappings effectively. The number of filters in each block would remain constant within a stage but may vary across stages to enable hierarchical feature learning. The existence of the shortcut connection helps mitigate issues such as vanishing gradients, enabling the training of very deep networks with efficiency and accuracy.

Besides the residual blocks, ResNet also contains other components. Typically, ResNet would begin with an initial convolutional layer with a 7×7 kernel, 64 filters, and a stride of 2. This initial layer provides feature extraction from the original input image. It is then followed by batch normalization and ReLU activation for stable and efficient training. A max pooling with a 3×3 kernel and a stride of 2 is applied to down-sample spatial dimensions while preserving essential features. After the initial convolutional layer, several residual blocks are connected to the network. The first convolutional layer in some residual blocks would adopt a stride of 2 to further down-sample the input by half. In such cases, the corresponding shortcut connection applies either padding with zero entries or linear projection on the input matrix to match the dimensions (shown in Fig. 3.2 as the shortcut connection with dotted arrow). After the initial convolutional layer and all the residual blocks, ResNet incorporates global average pooling for spatial aggregation and feature summarization. Finally, a fully connected layer with softmax activation is appended to ResNet to generates class probabilities. The number of neurons in the layer corresponds to the number of classes in the classification task. Therefore in general, a ResNet would contain an initial convolutional layer, several residual blocks, and a fully connected layer.

The exact number of residual blocks in ResNet can vary, yielding various complexities and capabilities of models. In ResNet18, '18' refers to the total number of the weight layers in the network. Therefore, ResNet18 would have 1 initial convolutional layer, 8 residual blocks (2 convolutional layers for each), and 1 fully connected layer,

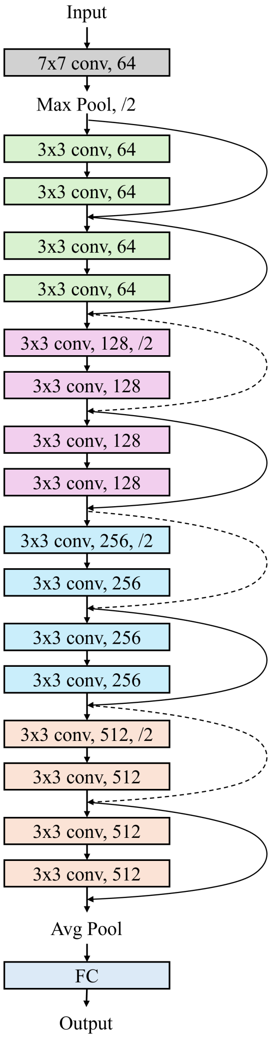


Figure 3.2: Structure of ResNet18

resulting in a total of 18 layers (see Fig. 3.2). Similarly, ResNet34, ResNet50, etc would have 34 layers, 50 layers, and more, respectively.

3.4.2 Visual Attention-Prompted Learning

The visual attention-prompted learning model (VAPL) [28] utilizes ResNet18 structure as the backbone of the model and adopts a parameter-sharing and co-activation framework [28] (Fig. 3.3). This framework mainly consists of two paralleled ResNet18 models denoted as f_m and f_o. Specifically, f_m is responsible for attention-prompted

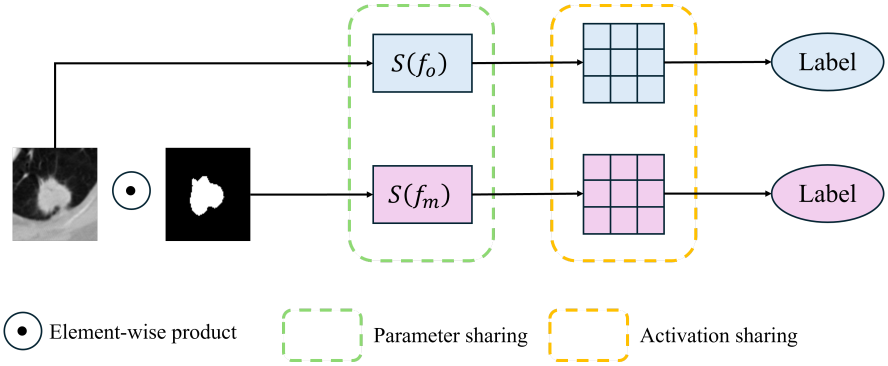


Figure 3.3: Structure of Visual Attention-Prompted Learning

prediction while f_o is responsible for prediction based solely on original input image.

The novelty of VAPL is that instead of training f_m and f_o independently, VAPL aims to leverage the interrelation of the two functions and to train both functions together, resulting in similar activation patterns within the two functions. The similar activation patterns would then enable the transfer of knowledge learned through visual prompt from f_m to f_o. This is achieved through the integration of two regularization terms: parameter-sharing regularization and co-activation regularization:

$$\mathcal{L}_{param}(\theta_{f_m}, \theta_{f_o}) = \| W_{f_o} - W_{f_m} \|_F^2 \tag{3.1}$$

$$\mathcal{L}_{Activ}(\theta_{f_m}, \theta_{f_o}) = \| S(f_o(X)) - S(f_m(X \odot P)) \|_F^2 \tag{3.2}$$

where $\| \cdot \|_F^2$ is the squared Frobenius norm [17], W_{f_o} and W_{f_m} represent the convolutional layer parameters of the two models, and $S(f_o(X))$ and $S(f_m(X \odot P))$ are the outputs of the two models before their fully connected layers. In addition, the cross-entropy loss between the ground truth label and the predictions made by the two models can be formulated as:

$$\mathcal{L}_{Pred} = -\sum_{i=1}^{K}\sum_{a=1}^{C} \hat{y}_{ia}(\log(p_m^{ia}) + \log(p_o^{ia})) \tag{3.3}$$

where $\hat{y}_{ia}$ is the ground truth label for class a of the i-th sample in one-hot encoding, and p_m^{ia} and p_o^{ia} are the predicted probabilities for class a of i-th sample made by f_m with visual prompt and by f_o without visual prompt respectively. Thus, the learning objective of VAPL can be formulated as:

$$\text{minimize} \quad \mathcal{L}_{Pred} + \lambda_1 \mathcal{L}_{Param} + \lambda_2 \mathcal{L}_{Activ} \tag{3.4}$$

where λ_1 and λ_2 are the weighting hyperparameters for parameter-sharing and co-activation.

Chapter 4

Experiments

This chapter demonstrates the design of the experiments, including the introduction of the datasets and explanation of the experimental setup. The design of experiments follows the same guideline as that of Zhang et al. [28].

4.1 Dataset

To test the effectiveness of the default model (VAPL) provided by the framework, experiments were conducted on four datasets: two sourced from MS COCO [15] and Places365 [30], respectively, and formulated into real-world scenarios, and two from the medical field - LIDC-IDRI (LIDC) [2] and the Pancreas dataset [23].

The dataset extracted from MS COCO was formulated into **Gender** classification [7]. $1,600$ images with 'men' or 'women' included in the captions were extracted from MS COCO. Images with both genders, multiple people, or unclear figures were excluded in the extraction process. Similarly, balanced 'nature' and 'urban' categories of images were extracted from Places365 for **Scene** Recognition task [7]. Each image from the both datasets were manually labeled for human-explanation annotation. From each dataset, only 100 images were randomly chosen for training to simulate the limited access to explanations.

As for the datasets in medical field, the **LIDC** dataset [28] contains lung computed tomography (CT) scans of nodules with annotated lesions. These scans were preprocessed into 2D images with size of 224 × 224 and augmented with noise to simulate incomplete prompts. Furthermore, negative samples were created by cropping surrounding areas without nodule from the original CT scans. The final extracted dataset contained 2625 positive samples with nodules and 65505 negative samples with no nodule and were randomly split into 100/1200/1200 for training, validation, and testing with the aim of simulating limited access to human explanations. Last but not least, the **Pancreas** dataset [28], with normal abdominal CT scans from Cancer Imaging Archive and abnormal CT scans from MSD, was preprocessed in a way similar to that of LIDC, resulting in 281 positive samples (with tumor) and 80 negative samples (without tumor). The samples were split into 30/30/rest while maintaining class balance for training, validation, and testing.

4.2 Experimental Setup

4.2.1 Comparison Methods

To evaluate the effectiveness of the default model (VAPL) provided by our framework, comparative studies were conducted with four top attention-guided learning methods - GRAIDA [6], HAICS [25], RES-G, and RES-L [7]. These comparison methods were implemented and trained while following the implementation guidelines presented in their papers. Furthermore, a ResNet18 [10], taking original images as input without any visual attention prompt, was adopted as the baseline model.

4.2.2 Implementation Details

To ensure the performances of all tested models were comparable and informative, the settings and hyperparameters for the training phase were set to be consistent among all models [28]. All tested models incorporated the ResNet18 architecture as the backbone. During the training, a batch size of 16 was uniformly adopted. The number of perturbed masks was set to be 5000 with a pixel conversion probability of 0.1. The training phase of each model contained a total of 10 epochs, and each epoch included 5 iterations, resulting in 50 training epochs in total. The Adam optimization [13] was applied to all the models with a learning rate of 0.0001. Finally, to evaluate the performances of each model, conventional classification metrics including accuracy, precision, recall, and the F1 score were applied.

Chapter 5

Analysis

This chapter offers a comprehensive analysis of the experiment results [28], as well as a detailed instruction on how to use our Visual Attention-Prompted Prediction Framework.

5.1 Results

Table 5.1 shows the performances of each tested methods in terms of the evaluation metrics on the LIDC and Pancreas datasets [28]. In general, our proposed method (VAPL) outperforms other attention-guided learning methods on both LIDC and Pancreas datasets. Specifically, our model achieves the best accuracy and F1 and the second best precision and recall in pulmonary nodule classification of LIDC, and the best accuracy, precision, recall, and F1 in pancreatic tumor classification of Pancreas. Furthermore, our method demonstrates significant improvements from the baseline ResNet18 method. With an accuracy of 69.45% and 92.31% and an F1 score of 71.13% and 95.30% on the two datasets respectively, our method shows respective enhancements of 3.05% and 7.22% on accuracy, and of 7.66% and 5.08% on F1 over the baseline ResNet18 approach. The significant improvements over the baseline represents the effectiveness of incorporating explanations into the prediction phase.

Table 5.1: Performances of tested models on LIDC and Pancreas datasets. The best result of each metric is in bold, and the second best result is underlined [28].

Model	LIDC				Pancreas			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Baseline	66.40	59.29	69.02	63.47	85.09	98.82	83.69	90.22
GRADIA	67.44	65.65	73.19	68.99	83.13	99.04	81.12	89.10
HAICS	66.86	64.71	74.60	69.14	86.44	98.99	85.10	91.24
RES-G	68.56	**67.93**	71.46	69.27	89.89	98.94	89.17	93.79
RES-L	68.35	65.97	**76.28**	70.55	89.79	98.07	89.88	93.78
VAPL	**69.45**	67.43	75.25	**71.13**	**92.31**	**99.84**	**91.03**	**95.30**

Meanwhile, the fact that our method achieves the highest F1 scores in both LIDC and Pancreas datasets proves that the overall performances of our method exceeds that of all other attention-guided learning methods.

On the other hand, table 5.2 exhibits the performances of all tested models on gender classification and scene recognition tasks in Gender and Scene datasets [28]. Similar to the results in LIDC and Pancreas (table 5.1), VAPL also demonstrates superior performances over other attention-guided learning models on both tasks by achieving the best accuracy, precision, and F1 scores in both datasets. VAPL also shows siginificant improvements from the baseline ResNet18 method. These results in Gender and Scene match with the results in LIDC and Pancreas, and thus further fortify the conclusions that our proposed way of utilizing attention prompt in prediction is effective and that our method outperforms other prominent attention-guided learning methods.

5.2 Instruction for the Visual Attention-Prompted Prediction Framework

This section provides a detailed step-by-step instruction on how to use our Visual Attention-Prompted Prediction Framework.

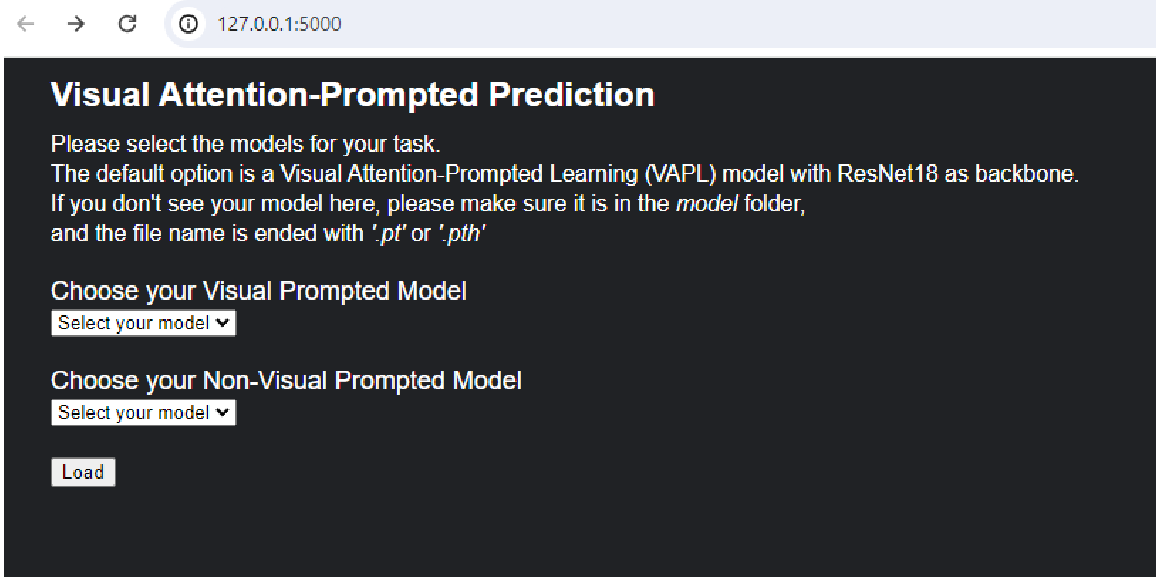
Table 5.2: Performances of tested models on Gender and Scene datasets. The best result of each metric is in bold, and the second best result is underlined [28].

Model	Gender				Scene			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Baseline	68.35	67.45	69.98	68.69	93.42	94.87	91.68	93.25
GRADIA	70.01	67.83	74.35	70.94	95.03	96.21	92.55	94.34
HAICS	69.29	66.42	73.61	69.83	94.89	95.73	92.94	94.31
RES-G	71.33	69.98	**78.53**	74.01	95.91	96.22	**95.35**	95.78
RES-L	70.39	68.41	73.29	70.77	95.53	96.98	94.56	95.75
VAPL	**73.36**	**71.43**	76.88	**74.05**	**96.39**	**97.43**	94.48	**95.93**

After obtaining the framework from GitHub repository and configuring the environment with ’*requirements.txt*’, the user needs to put the model and dataset into the corresponding directories. The model provided by the user should be a PyTorch model file with a ’*.pt*’ or ’*.pth*’ file extension, and it should go into the ’*./model/*’ directory. Meanwhile, the folder that contains images for labeling and prediction should go into the ’*./static/images/*’ directory.

Then, the user needs to open a terminal, initiate the corresponding virtual environment, and navigate to the directory containing the framework. Before running the framework, the user would need to run the ’*init_code.py*’ script first. This process scans all the images uploaded by the user and prepare them for the annotation task. This process also enables the framework’s ability to resume from where the user left out last time.

Now, the user can start labeling images for visual attention-prompted prediction by running ’*app.py*’ and going to `http://127.0.0.1:5000` with any web browser. The first page that the user would see is the page for model selection (Fig. 5.1. The framework requires two models in total: a model to predict with visual prompt and a model to predict without visual prompt. However, the same model can be used, and is encouraged to be used, for both tasks, since this enables the direct comparison between the two tasks and reflects the effectiveness of visual prompt. The user can click down the drop-down menu and select a model from there (Fig. 5.2). The framework would automatically filter out files in the '*./model/*' directory with extensions other than '*.pt*' or '*.pth*'. After a desired model is selected for both tracks, the user can move on to the next stage by clicking 'load'.



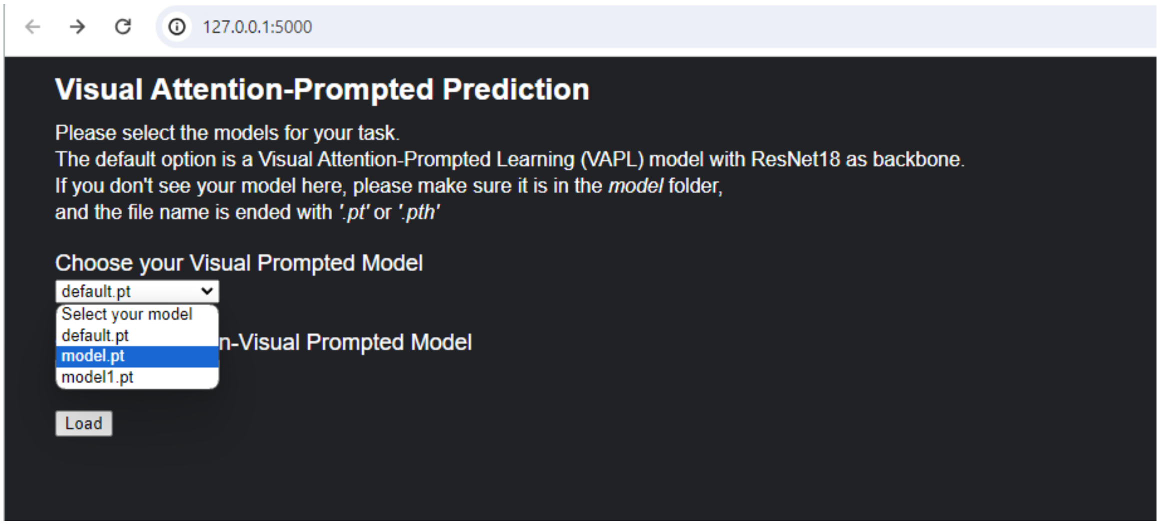
Visual Attention-Prompted Prediction

Please select the models for your task.
The default option is a Visual Attention-Prompted Learning (VAPL) model with ResNet18 as backbone.
If you don't see your model here, please make sure it is in the *model* folder,
and the file name is ended with '*.pt*' or '*.pth*'

Choose your Visual Prompted Model

Choose your Non-Visual Prompted Model

Figure 5.1: Framework Page for Model Selection



Visual Attention-Prompted Prediction

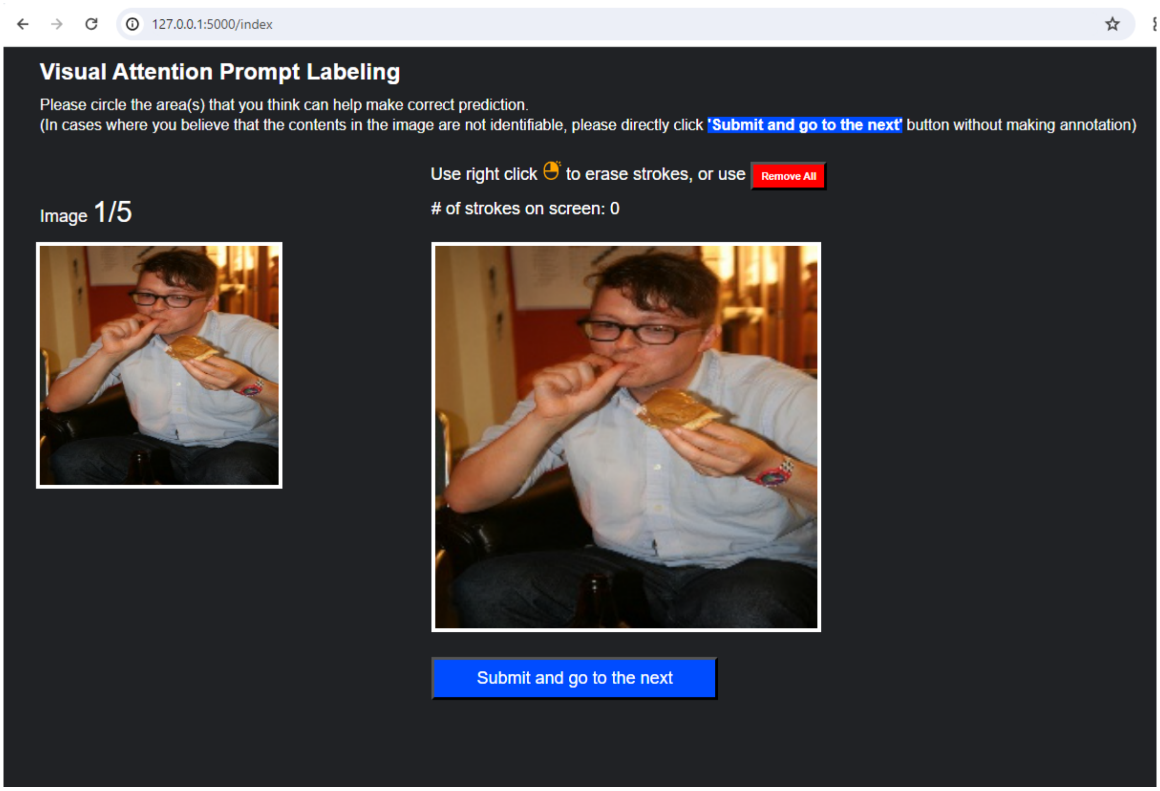
Please select the models for your task.
The default option is a Visual Attention-Prompted Learning (VAPL) model with ResNet18 as backbone.
If you don't see your model here, please make sure it is in the *model* folder,
and the file name is ended with '*.pt*' or '*.pth*'

Choose your Visual Prompted Model

n-Visual Prompted Model

Figure 5.2: Drop-down menu of Model Selection

This new stage features one of the images provided by the user on the side of the page as a reference and a canvas for the user to draw their explanations on the



Visual Attention Prompt Labeling

Please circle the area(s) that you think can help make correct prediction.
(In cases where you believe that the contents in the image are not identifiable, please directly click 'Submit and go to the next' button without making annotation)

Use right click to erase strokes, or use Remove All

# of strokes on screen: 0

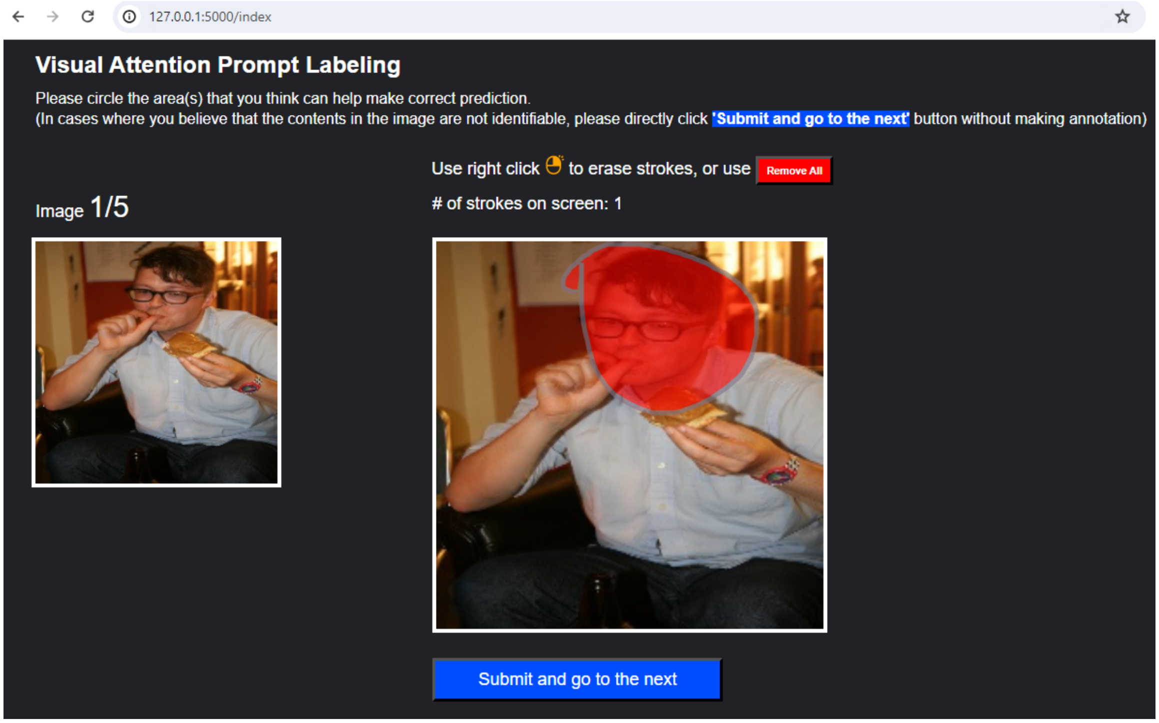
Image 1/5

Figure 5.3: Framework Page for Human Explanation Annotation

original image (Fig. 5.3). The user can use left click to draw circles on the canvas, highlighting areas deemed to be important for prediction (Fig. 5.4). If the user is not satisfied with the drawing, he/she can use the right click to erase the drawing stroke by stroke or click 'Remove All' to erase all existing drawing. Once the user has a drawing that he/she thinks reasonable, he/she can click 'Submit' to send the image and the annotation into the models for prediction.

Once the two models have generated outputs based on the input image and human labeled explanation, the framework would convert models' output from probabilities of each class to the class label with the highest probability (Fig. 5.5). Along with the prediction results, the original input image and highlighted areas would also be presented for examination and comparison. User can click 'next' to continue annotating other images. This would take the user back to the page similar to Fig. 5.3.

When all the images provided by the user have been annotated, an ending page is presented to the user (Fig. 5.6). The framework would direct the user to a specific folder where all the user's annotations are stored. The user can choose to extract them for future analysis.



Visual Attention Prompt Labeling

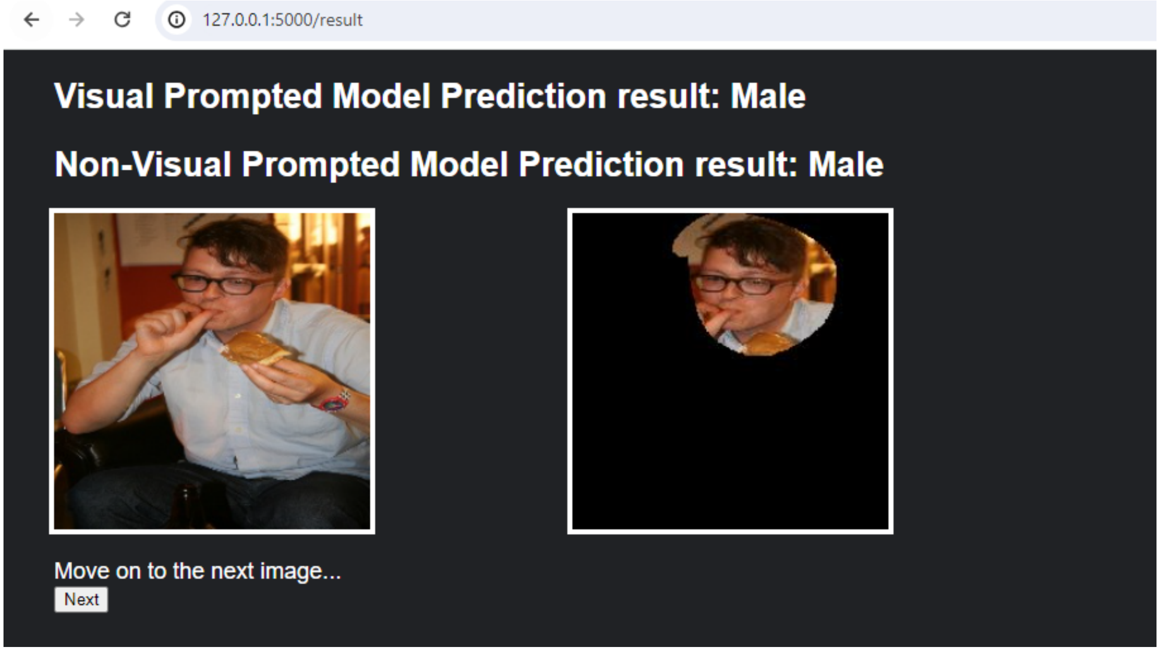
Please circle the area(s) that you think can help make correct prediction.
(In cases where you believe that the contents in the image are not identifiable, please directly click 'Submit and go to the next' button without making annotation)

Use right click to erase strokes, or use Remove All

# of strokes on screen: 1

Image 1/5

Figure 5.4: Ongoing Annotation

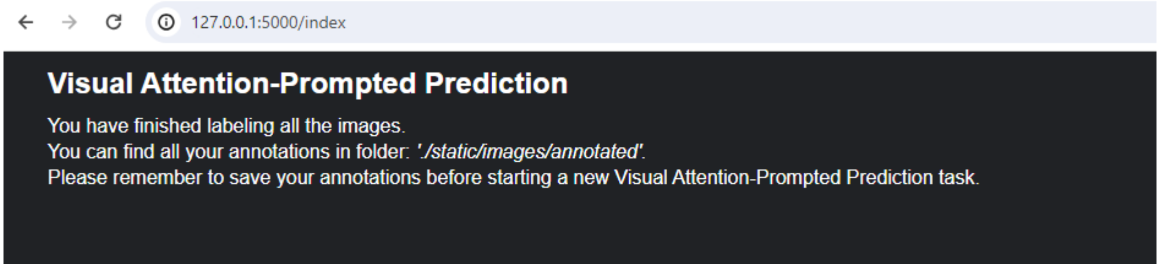


Visual Prompted Model Prediction result: Male

Non-Visual Prompted Model Prediction result: Male

Move on to the next image...

Figure 5.5: Framework Page for Prediction Result



Visual Attention-Prompted Prediction

You have finished labeling all the images.
You can find all your annotations in folder: *'./static/images/annotated'*.
Please remember to save your annotations before starting a new Visual Attention-Prompted Prediction task.

Figure 5.6: Framework Ending Page

Chapter 6

Conclusion

This research presents a novel visual attention-prompted prediction framework, which incorporates visual attention prompts into the CNN model's decision-making process. The framework effectively addresses challenges in visual attention-prompted prediction, particularly the issue of insufficient prompts for learning, by enabling real time visual prompts annotation for prompted prediction. The framework also provides a state of the art visual attention-prompted learning model that enables knowledge transfer from prompted model to non-prompted model. Meanwhile, comprehensive experiments have been done over the default model for the framework and other visual attention-guided learning methods using four distinct real-world datasets, demonstrating the effectiveness of visual explanations in improving model's predictive power and showcasing the outstanding performances of the adopted model.

Bibliography

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, PP:1–1, 09 2018. doi: 10.1109/ACCESS.2018.2870052.

[2] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.

[3] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations, 2018.

[4] Shi Chen, Ming Jiang, Jinhui Yang, and Qi Zhao. Air: Attention with reasoning capability, 2020.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[6] Yuyang Gao, Tong Sun, Liang Zhao, and Sungsoo Hong. Aligning eyes between humans and deep neural network through interactive attention alignment, 2022.

[7] Yuyang Gao, Tong Steven Sun, Guangji Bai, Siyi Gu, Sungsoo Ray Hong, and Zhao Liang. Res: A robust framework for guiding visual explanation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. ACM, August 2022. doi: 10.1145/3534678.3539419. URL `http://dx.doi.org/10.1145/3534678.3539419`.

[8] Siyi Gu, Yifei Zhang, Yuyang Gao, Xiaofeng Yang, and Liang Zhao. Essa: Explanation iterative supervision via saliency-guided data augmentation. pages 567–576, 08 2023. doi: 10.1145/3580305.3599336.

[9] Daniel Hajialigol, Hanwen Liu, and Xuan Wang. Xai-class: Explanation-enhanced text classification with extremely weak supervision, 2023.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[11] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023.

[12] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning, 2022.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[14] H. Li, D. Zhang, N. Liu, L. Cheng, Y. Dai, C. Zhang, X. Wang, and J. Han. Boosting low-data instance segmentation by unsupervised pre-training with saliency prompt. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15485–15494, Los Alamitos, CA, USA, jun

2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01486. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01486.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[16] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), jan 2023. ISSN 0360-0300. doi: 10.1145/3560815. URL https://doi.org/10.1145/3560815.

[17] Changxue Ma, Yves Kamp, and Lei F. Willems. A frobenius norm approach to glottal closure detection from the speech signal. *IEEE Trans. Speech Audio Process.*, 2:258–265, 1994. URL https://api.semanticscholar.org/CorpusID:22087919.

[18] Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. Wt5?! training text-to-text models to explain their predictions, 2020.

[19] Roni Paiss, Hila Chefer, and Lior Wolf. No token left behind: Explainability-aided image classification and generation, 2022.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning

library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[21] Zhongang Qi, Saeed Khorram, and Li Fuxin. Visualizing deep networks by optimizing with integrated gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11890–11898, April 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i07.6863. URL http://dx.doi.org/10.1609/aaai.v34i07.6863.

[22] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[23] Holger R. Roth, Le Lu, Amal Farag, Hoo-Chang Shin, Jiamin Liu, Evrim Turkbey, and Ronald M. Summers. Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation, 2015.

[24] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

[25] Haifeng Shen, Kewen Liao, Zhibin Liao, Job Doornberg, Maoying Qiao, Anton Hengel, and Johan Verjans. Human-ai interactive and continuous sensemaking: A case study of image classification using scribble attention maps. pages 1–8, 05 2021. doi: 10.1145/3411763.3451798.

[26] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning*

systems, 32(11):4793—4813, November 2021. ISSN 2162-237X. doi: 10.1109/tnnls.2020.3027314. URL `https://ieeexplore.ieee.org/ielx7/5962385/9591206/09233366.pdf`.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[28] Yifei Zhang, Siyi Gu, Bo Pan, Guangji Bai, Xiaofeng Yang, and Liang Zhao. Visual attention-prompted prediction and learning, 2023.

[29] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization, 2015.

[30] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.