**Abstract**

Convolutional neural network (CNN) models have seen advanced improvements in performance in various domains, but lack of interpretability is a major barrier to assurance and regulation during operation for acceptance and deployment of AI-assisted applications. There have been many works on input interpretability focusing on analyzing the input-output relations, but the internal logic of models has not been clarified in the current mainstream interpretability methods. In this study, we propose a novel hybrid CNN-interpreter through: (1) An original forward propagation mechanism to examine the layer-specific prediction results for local interpretability. (2) A new global interpretability that indicates the feature correlation and filter importance effects. By combining the local and global interpretabilities, hybrid CNN-interpreter enables us to have a solid understanding and monitoring of model context during the whole learning process with detailed and consistent representations. Finally, the proposed interpretabilities have been demonstrated to adapt to various CNN-based model structures.

**Keywords** Hybrid CNN-interpreter, local interpretability, global interpretability, correlation, filter importance

1 Introduction

Although the performance of convolutional neural network (CNN) models has significantly increased over the past decade, yet without reliable interpretabilities that effectively represent the learning processes, humans still consider...
Figure 1: Hybrid CNN-interpreter: understand how different convolutional neural networks learn through layers and filters: 1) Local interpretability to represent each layer’s output and learning ability; 2) Global interpretability to explore the representations ability of feature maps (learned by different filters) by using regression models; 3) feature importance analytic to indicate the contributions of convolution filters when making predictions. Overall, it enables for both local and global interpretability for different CNN structures.
Hybrid CNN-Interpreter

We present a novel hybrid CNN interpreter, which builds stacking ensemble models of each layer for local interpretabilities and extends local interpretabilities to compute global correlation to assess the importance of convolution filters. The results are illustrated by binary classification of brain tumor data in the different model structures. It makes three innovative improvements:

1. Build an original stacking forward propagation algorithm by computing the contributions of each layer to the final prediction. In stacking, each time takes the outputs of the specific layer as input and connects to the final probability mapping layer directly, which can use the set of predictions as a local context and examine feature maps learned by different layers’ ability for the final prediction contributions.

2. Extend local interpretabilities to a global interpretation by layer-based regression models, which are constructed by using all the local interpretabilities as dependent variables and each feature map representation as an independent variable across the entire dataset. This enables the examination of the representation ability of feature maps (learned by different filters) for interpreting a model’s global behavior.

3. Extract filter importance by assigning scores to each filter in each layer of a model that indicates the correlation of convolution filters when making a prediction, which can help people understand the relationship between the filters and the target variable and can conditionally improve the performance of the model, potentially making the model lighter and speeding up the model’s working by removing the unimportant filters.

4. Applicable for a broad range of CNN-based models, which can be utilized to interpret both sequence and nonsequence-focused models. For the sequence models, the interpretabilities can be through each layer to provide the learning context, whereas for the nonsequence models, the interpretabilities can be customized for different stages, blocks, or specific layers, etc. This reveals how different learning is in different models and in different training processes.

2 Related Work

2.1 Convolutional neural network structures

CNNs with strong representation ability of deep structures have ever-increasing popularity in many applications. Figure 1 shows the historical evolution of various CNN models. AlexNet [8] is a leading structure of convolutional neural networks and has huge applications for classification tasks. The evolution after the AlexNet can be mainly summarized in two ways: sequence-focused models by increasing the depth of networks and nonsequence-focused models by adding units or modules.

Figure 2: Summarization of CNN structures

Sequence focused models: by leveraging the sequence information through layers to improve the model performance, such as VGG16 and VGG19 [9], MSRANet [10]. All of these models are focused on using multiple convolutional

3
2.2 Visual interpretabilities of CNN-based models

According to different implementation methods, the visual interpretability methods can be divided into three different categories: input visualization method, model visualization method, and mixed visualization method. Table 1 states the overall summary of possible explainable methods.

The input visualization explains processes simultaneously from the initial input stage to the final output result. For example, Jeyaraj and Nadar [14] stated a regression-based partitioned method for oral cancer diagnosis. The network was demonstrated with two partitioned layers for labelling and categorized a multidimensional hyperspectral picture by tagging the region of interest. It handled feature maps with little variation and complex vector feature maps. Another example was the importance estimation network produced by Gu et al. [15], which was diagnosed by the classification network by investigating the irrelevant information. The model aims to detect the most significant sections of the original input images and provided an accurate diagnosis after being trained with the proper regularization settings. The input visualization merely provides a user-friendly interface for seeing and understanding input data, but no information on how the relevant features contribute to the prediction is supplied.

Generally, model visualization is based on the level of layers and finds out the prediction results of the model between different layers. For example, Graziani et al. [16] employed “The concept activation vector (TCAV)” to transform medical pictures into quantitative characteristics by using radiomics, which focused on explaining the predicted output globally according to high-level visual features. Additionally, Villain et al. stated a novel GradCAM method on brain MRI image datasets [17]. To identify the regions of interest by the visual convolutional neural network models, the authors claimed that the output of every convolution layer was collected by the global average pool layer and merged to obtain a single activation map to visualize. The model visualization could find hidden problems inside the model, but the currently selected features used for model interpretability may not be repeatable in each input.

Hybrid visualization model refers to a combination of both input visualization and model visualization. This model tries to focus on not only explaining the local prediction but also exploring global knowledge inside the model. Hybrid visualization model concentrates on finding the connections or correlations between the input aspect and model layer aspect. Unfortunately, to my knowledge, there are no obvious research on mixed visualization method.

In this paper, we discuss how to interpret CNN-based models using hybrid visualization to make the interpretability appropriate for different model structures. We describe the local interpretability by building a set of stacked ensemble models to provide various input visualization. Then, we extend local interpretability to model visualization by using regression-based analytics. Finally, we identify the importance of each filter in each layer to represent the global context of the models.
3 Method

Hybrid CNN-interpreter aims to explain the deep learning model’s different layers and the filter’s feature representation ability for the final prediction contribution. It provides global insight into not only the model’s layer level, but also the filter level along that layer. The output of the hybrid CNN-interpreter is a layer and filter-based importance distribution matrix, representing the importance level of feature maps learned from different layers and filters by the model. The hybrid CNN-interpreter consists of the original CNN-model forward propagation module, linear regression module and filter importance analysis module.

![Diagram of Hybrid CNN-interpreter framework](image)

Figure 3: Hybrid CNN-interpreter framework: including Original CNN-model Forward Propagation, Linear Regression Module and Filter Importance Analysis.

3.1 Stacking Forward Propagation

In the first original forward propagation module, each image is sent to the original model. We skip and connect each layer \( l_i(X_{ij}) \) directly to the final probability mapping layer \( f(\cdot) \), and the output probability \( Y_i \) is recorded. This is to examine feature maps learned by different layers’ ability for the final classification contribution (Equation 1).

\[
f[l_i(X_{ij})] = Y_i
\]

where \( X_{ij} \) represents the \( j^{th} \) feature map at the \( i^{th} \) layer.

3.2 Linear Regression Module

In the second step, a linear regression model is constructed by using the output probability as a dependent variable and each feature map’s mean value as independent variables. This is to examine feature maps’ (learned by different filters) representation ability for the classification (Equation 2).

\[
Y_i = b_{i0} + \sum b_{ij}X_{ij}
\]
where $X_{ij}$ represents the mean value of feature map $X_{ij}$. We applied L2 regularization and the objective function is represented in Equation 3:

$$L = \sum (\hat{Y}_i - Y_i)^2 + \lambda \sum b_{ij}^2$$  \hspace{1cm} (3)$$

where $\hat{Y}_i$ is the predicted value and $Y_i$ is the true value.

Algorithm 1 indicates the process of how $X_{ij}$ are extracted from the model outputs.

---

**Algorithm 1** Mean values of feature maps extraction

1: procedure EXPORT-FEATURES(img,lnames) ⊲ The input of this algorithm are images and the layer names of the model
2: model.predict(img) ← features ⊲ The feature maps are obtained by visualisation the model prediction.
3: for lname,features do
4:   mean-list = []
5:   for i in range(features-number) do
6:     x = features[0, :, :, i]  ⊲ x.mean() the mathematical mean value of x
7:     x = x - x.mean()  ⊲ x.std() the mathematical standard deviation value of x
8:     x = x / x.std()  ⊲ Use numpy.clip() function to limit values outside the interval are clipped to the interval edges
9:     x = x + 64
10:    x = numpy.clip(x, 0, 255)
11:    mean-list.append(x)  ⊲ Store the mean value of the feature maps into mean-list
12:  end for
13: end for
14: return mean-list  ⊲ The output of the algorithm is a list with the mean value of the feature maps
15: end procedure

In our experiment, we use Ridge regression model [18] to calculate the variance of our feature maps. The Ridge model solves regression problems with an $L_2$ regularization loss function, the equation of the loss is shown in Equation 3. In the Ridge model, the parameter $\alpha$ ($\alpha = 1$ in our experiment) is used to control the regularization strength. The input of our regression model is the mathematical mean value of our feature maps and the output of the CNN models after prediction (confidence score in our experiment). Algorithm 2 shows the details of

**Algorithm 2** Regression Algorithm

1: procedure REGRESSION(layer-names,f-mean,CNN-predictions) ⊲ The input of this algorithm are mean values of the feature maps and the layer names of the model
2:   x = f-mean
3:   y = CNN-predictions
4:   model = Ridge(alpha=1.0)  ⊲ The model is loaded from sklearn library
5:   predictions = model.predict(x)
6:   coefficients, standard-errors, t-values, p-values = sklearn-math-function(y, predictions,features)  ⊲ we use a combination of mathematical tools from sklearn library to calculate coefficients, standard-errors, t-values and p-values
7:   return coefficients, standard-errors, t-values, p-values
8: end procedure

Then, the regression model will return four parameters to represent the importance of our feature map, i.e. the coefficient of determination ($R^2$), standard-errors (SEs), t-values and p-values. The equation of $R^2$ is demonstrated in 4. The meaning of $R^2$ is the proportion of the variance of the variable that is explained by this predicted model. In our experiment, we use $R^2$ to evaluate the reliability of the prediction, if the value of $R^2$ close to 1, which means the model predictions are reliable and effective. Conversely, if the value of $R^2$ is close to 0, the predictions of the model are not reliable and ineffective.

$$R^2 = \frac{\sum_i e_i^2}{\sum_i (y_i - \overline{y})^2}$$  \hspace{1cm} (4)$$

The equation of SE is shown in Equation 5. The SE is another statistical tool for us to evaluate the degree of deviation of the sample mean from the overall true mean. Normally, a higher SE value indicates that the mean value of the feature
Hybrid CNN-Interpreter

maps is more distinct, whereas a smaller SE value means the features extracted are similar.

\[ SE = \frac{\sigma}{\sqrt{n}} \]  

(5)

where \( SE \) is the standard error of the sample, \( \sigma \) is the sample standard deviation and \( n \) is the number of samples.

3.3 Filter Importance Analysis Module

In the third filter importance analysis module, a matrix of different filters’ representation scores has been calculated, the matrix helps people understand the contribution of the feature map learned by different filters in each layer of the network to the final classification result.

In our settings, the t-value (Equation 6) is used to indicate the level of difference between features. The t-value is calculated as the difference expressed in standard error units. It means that the standard deviation estimated is far away from 0. A large t-value indicates the evidence against the null hypothesis, in other words, we could declare a relationship between the CNN-predictions and the feature maps.

\[ t = \frac{\bar{x} - \mu_0}{s \sqrt{n}} \]  

(6)

where \( \bar{x} \) is the sample mean, \( \mu_0 \) is the population mean, \( s \) is the sample standard deviation and \( n \) is sample size.

We also used p-value (Equation 7) to evaluate two variables, namely, they are the importance of the features and the CNN model prediction accuracy. We want to examine the relationship between the predictor variables and the response variable to find out if higher accuracy in model prediction will result in higher importance values in the feature maps’ mean values.

\[ p = \frac{\hat{p} - p_0}{\sqrt{p_0 (1 - p_0)}} \]  

(7)

where \( \hat{p} \) is sample proportion, \( p_0 \) is assumed population proportion in the null hypothesis and \( n \) is the sample size.

4 Experiment and Discussion

4.1 Data preparation

A total of 253 public brain MRI images dataset (155 tumorous images and 98 nontumorous images) are used to build the classification models. Since CNNs can be independent of translation, view position, size, and lighting [19], the data augmentation was applied by manually flipping and rotating the image sets. Moreover, considering the dimension of the image may be changed after rotation [20], the image will be only rotated arbitrarily between 0 and 10 degrees in our experiments.

Finally, after the above data augmentation, the dataset has 1,085 positive and 980 negative examples. We use 70% of the images for training and 15% of the images for validating to generate the AlexNet and ResNet classifier models. The remaining of 15% of the images are for model interpretability in our proposed hybrid CNN-interpreter.

4.2 Model selection

In this study, two alternative CNNs, namely, AlexNet and ResNet, are selected to demonstrate the detailed and consistent interpretabilities of our proposed methods. The model structures of AlexNet and ResNet are represented in Table 1 and 3 respectively.

4.3 Local interpretability for CNN-based Models

The hybrid CNN-interpreter enables local interpretability with individual repeatable capacity by building a set of stacked ensemble submodels. Each submodel is developed by connecting the output of each layer in the original model and the final probability mapping layer to output prediction results, representing the model’s learning ability. For the AlexNet model, the local interpretability will focus on each layer. For the ResNet model, the local interpretability will concentrate on each stage and component in each stage. The hybrid CNN interpreter will collect this set of prediction results for individual predictions.
Table 2: The network structure of AlexNet model

| Layer type          | Size of output feature map | Number of filters |
|---------------------|-----------------------------|-------------------|
| Conv_2d             | 59*59                       | 64                |
| Bn                  | 59*59                       | 64                |
| Max_pooling_2d      | 29*29                       | 64                |
| Conv_2d_1           | 29*29                       | 64                |
| Bn_1                | 29*29                       | 64                |
| Max_pooling_2d_1    | 14*14                       | 64                |
| Conv_2d_2           | 14*14                       | 64                |
| Bn_2                | 14*14                       | 64                |
| Conv_2d_3           | 14*14                       | 64                |
| Bn_3                | 14*14                       | 64                |
| Conv_2d_4           | 14*14                       | 64                |
| Bn_4                | 14*14                       | 64                |
| Max_pooling_2d_2    | 7*7                         | 64                |

Table 3: The network structure of ResNet model

| Stage    | Operations                         | Size of output feature map | Number of filters |
|----------|------------------------------------|-----------------------------|-------------------|
| Stage 1  | Conv_1                             | 120*120                     | 64                |
|          | Bn_1                               | 120*120                     | 64                |
|          | Activation                          | 120*120                     | 64                |
|          | Max_pooling_2d                      | 59*59                       | 64                |
| Stage 2  | Stage2_input                        | 59*59                       | 64                |
|          | Convolutional block                 | 59*59                       | 256               |
|          | Identity block 1                    | 59*59                       | 256               |
|          | Identity block 2                    | 59*59                       | 256               |
| Stage 3  | Stage3_input                        | 30*30                       | 128               |
|          | Convolutional block                 | 30*30                       | 512               |
|          | Identity block 1                    | 30*30                       | 512               |
|          | Identity block 2                    | 30*30                       | 512               |
|          | Identity block 3                    | 30*30                       | 512               |
| Stage 4  | Stage4_input                        | 15*15                       | 256               |
|          | Convolutional block                 | 15*15                       | 1024              |
|          | Identity block 1                    | 15*15                       | 1024              |
|          | Identity block 2                    | 15*15                       | 1024              |
|          | Identity block 3                    | 15*15                       | 1024              |
|          | Identity block 4                    | 15*15                       | 1024              |
|          | Identity block 5                    | 15*15                       | 1024              |
| Stage 5  | Stage5_input                        | 8*8                         | 512               |
|          | Convolutional block                 | 8*8                         | 2048              |
|          | Identity block 1                    | 8*8                         | 2048              |
|          | Identity block 2                    | 8*8                         | 2048              |
In our experiments, four different samples are input into the AlexNet model to explore the local interpretability, where we pick up both tumour and nontumour images as well as both the final correct prediction and wrong prediction as shown in the Table 4.3.

| Figures | Tumor image | Nontumor image | Result |
|---------|-------------|----------------|--------|
| Figure 4(a) | ✓ | | ✗ |
| Figure 4(b) | ✓ | | ✓ |
| Figure 4(c) | | ✓ | ✗ |
| Figure 4(d) | ✓ | ✓ | ✓ |

Figure 4 represents the prediction results of each layer in the AlexNet model, which is guaranteed consistent for repeatable running. From the local interpretability, we can see that each layer of different samples makes different individual predictions. Moreover, it is not an actual rule "more layers = better performance". For example, the...
Hybrid CNN-Interpreter

Conv2d_1 layer demonstrates the best performance in Figure (a) and the Batch_normalization_2 layer achieved the better performance in Figure 4(c).

Additionally, we interpret the ResNet model by computing a set of prediction results for each stage and each block as well. As shown in Figure 5, the ResNet model shows that stages two and three have better performance compared with other stages in this test sample. For each internal stage, our proposed local interpretability can also provide a shortcut how learning by jumping over different residual blocks. For example, stage 5 is composed of one convolutional block and two identity blocks. The best performance in this stage was achieved by connecting the input layer and the first identity block.

Figure 5: Local interpretability results of ResNet Model: tumor image with final correct prediction results

4.4 Global interpretability for CNN-based Models

By combining the local interpretability of each layer across the entire training data, we present the regression-based methods to provide global insight into each filter in each layer, which will capture the global pattern of filters and the relationship between feature representations and prediction results. The experiments from the hybrid CNN-interpreter for global interpretability cover: 1) summary plots of the linear coefficient to represent the strength and direction of the linear relationship between each filter and prediction results in each layer, stage or internal block; 2) correlation analysis through layers of the AlexNet model and stages/blocks of the ResNet model respectively; 3) Filter importance analysis to reveal the details of positive and negative feature representations.

We summarize the linear coefficients of all layers in the AlexNet model as shown in Figure 6, and the distribution of each filter’s coefficient in each layer can also be displayed by highlighting the positive as blue and the negative as red. Based on the results, we can see that filters 20 and 31 show the majority of positive effects among all the layers, whereas filters 28 and 35 show negative effects in most layers.

We also get the linear coefficients of all stages and every internal block in the ResNet model. The overall distribution of coefficients among all stages can be shown in Figure 7(a), and by using stage 4 as a reference, the positive and negative coefficients can be displayed in Figure 7(b). Furthermore, to explore richer information, we can set a small range of filters such as filters 256-288 in stage 4, some consistent patterns among all the internal blocks can be revealed as shown in Figure 7(c), filter 263 is positive in most blocks, and filters 259, 270, 284, and 285 usually are negative in majority internal blocks. This type of information can provide evidence for deriving lightweight models by eliminating layers, stages, or filters.
Based on the linear coefficient results above, we analyze the correlation between layers in the AlexNet model and the correlation between stages and internal blocks in each stage. Figure 6(a), we shows that the pairwise layers such as Conv2d and Batch_normalization (Bn), Conv2d_1 and Batch_normalization_1(Bn_1), etc. always have strong correlation with each other, which can reflect the local interpretability that these pairwise layers get the similar prediction results as well. Moreover, we can clearly see the Conv2d_4 and Bn_4 layers have global negative effects for the final prediction results, compared with the local interpretability in Figure 4, the prediction results in the Max_pooling_2d_2 layer almost get noticeably different results compared with Conv2d_4 layer and Bn_4 layers, which can approve the consistency between local and global interpretability.

For the ResNet model, Figure 6(b) shows that stage 4 has negative effects from stage 1 to stage 3. This reflects the consistency with the local interpretability in Figure 5 that the performance increased from stage 1 to stage 3 but dropped starting from the stage 4. Moreover, the correlations among the internal blocks in stage 4 indicate a generally positive correlation with each residual block. However, the correlations get less and less along with the blocks getting deeper, such as the correlation between Conv_block and following identity blocks, is getting lower as shown in Figure 6(c).

Finally, we extract the most essential filters in each layer to reveal the details of positive and negative feature representations. Figure 7 shows each layer’s most significant five positives and five negatives using the same sample image in Figure 6(a). From Figure 7 we can see the different filters positively and negatively impact on different layers. For example, filter 20 positively impacts a total of nine layers that always focus on lower and middle layers, whereas filter 37 shows a positive impact on deeper layers. Filters 35 and 28 have a negative impact throughout the whole learning process. Additionally, the pairwise learning layers (pair of convolutional layer and batch normalization layer) show similar feature representations, which is consistent with our local interpretability. Figure 8 summarises the filter importance analysis for the ResNet model. We pick up the input layer and output layer for each stage as well as the output layer for each internal block to show the feature map representations and their linear coefficient. Since the ResNet model is much more deeper than the AlexNet model, we can see that the deeper the layer, the more ambiguous the feature maps and the more difficult it is for the humans to understand their semantic information.

By combining local interpretability and global interpretability, the experiments show that when the forward probability of the submodel is high, the features learned by the selected top five positive filters are consistent with human cognition with more significant semantic information, such as both the five filters (20,26,62,45,and 31) of Conv_2d_1 layer in AlexNet model and the filters (151,24,118,190,and 48) of identity_block_2 output layer in ResNet model showing the part of important features in the tumor area. Moreover, when the forward probability of the submodel is low, filters marked as negative effects can learn valuable semantic information. For example, the feature representation of
Figure 7: **Summary plot of linear coefficients in ResNet Model.**

- **a.** Linear coefficients of all stages.
- **b.** Positive and negative coefficient distribution in stage 4.
- **c.** Coefficient patterns of internal blocks in stage 4 by setting a specific filter range.
Figure 8: Correlation analysis among layers, stages or blocks for global interpretability. a. Correlation coefficients between layers in AlexNet Model. b. Correlation coefficients between stages in ResNet Model. c. Correlation coefficients between internal blocks in ResNet Model (by using stage 4 as reference).

5 Conclusion and Future Work

The hybrid CNN interpreter helps users to have a deep conceptual understanding of CNN-based models by providing both local and global interpretabilities. The local interpretability by using original forward propagation can reveal how image data progresses through the layers of the CNN-based models, whereas the global filter importance based on the linear regression module can indicate how much each filter contributes to the model prediction.

The hybrid CNN interpreter can be widely used in different computer vision tasks, such as classification, object detection, and segmentation. The proposed interpreter can be flexibly adapted to various CNNs (layer-based, stage-based, or internal-block-based). The correlations between layers, stages, or blocks can also be generalized to understand the context of complex CNN structures. By demonstrating how to apply the hybrid CNN interpreter to explain different types of CNN-based models, we take brain tumor classification tasks to provide an interpretation of the AlexNet and ResNet models. The experiment results showed the efficiency and consistency between local and global interpretabilities, as well as among different models.

The proposed interpreter can be used to improve and debug models and to make CNN-based models more accurate and reliable. The local interpreter can reveal prediction results for specific samples of different learning processes, and based on these interpretabilities, we can set the gating and memory mechanisms to debug models that can help us identify where to access that memory and where to ignore it during the whole learning process. For example, the model can skip some layers with bad performances and tell where to access and make connections again. The feature correlations and filter importance identified by the global interpreter enable developers to determine which filter can be eliminated to get better performance.

References

[1] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil K Jain, and Jiliang Tang. Trustworthy ai: A computational perspective. arXiv preprint arXiv:2107.06641, 2021.
[2] Yiming Zhang, Ying Weng, and Jonathan Lund. Applications of explainable artificial intelligence in diagnosis and surgery. *Diagnostics*, 12(2):237, 2022.

[3] Urja Pawar, Donna O’Shea, Susan Rea, and Ruairi O’Reilly. Explainable ai in healthcare. In *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, pages 1–2. IEEE, 2020.

[4] Shahin Atakishiyev, Mohammad Salameh, Hengshuai Yao, and Randy Goebel. Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. *arXiv preprint arXiv:2112.11561*, 2021.

[5] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[6] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[7] Kerelous Waghen and Mohamed-Salah Ouali. Multi-level interpretable logic tree analysis: A data-driven approach for hierarchical causality analysis. *Expert Systems with Applications*, 178:115035, 2021.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Donghao Gu, Yaowei Li, Feng Jiang, Zhaojing Wen, Shaohui Liu, Wuzhen Shi, Guangming Lu, and Changsheng Zhou. Vinet: A visually interpretable image diagnosis network. *IEEE Transactions on Multimedia*, 22(7):1720–1729, 2020.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Pandia Rajan Jeyaraj and Edward Rajan Samuel Nadar. Computer-assisted medical image classification for early diagnosis of oral cancer employing deep learning algorithm. *Journal of cancer research and clinical oncology*, 145(4):829–837, 2019.

[15] Donghao Gu, Yaowei Li, Feng Jiang, Zhaojing Wen, Shaohui Liu, Wuzhen Shi, Guangming Lu, and Changsheng Zhou. Vinet: A visually interpretable image diagnosis network. *IEEE Transactions on Multimedia*, 22(7):1720–1729, 2020.

[16] Mara Graziani, Vincent Andreczky, Stéphane Marchand-Maillet, and Henning Müller. Concept attribution: Explaining cnn decisions to physicians. *Computers in biology and medicine*, 123:103865, 2020.

[17] Edouard Villain, Giulia Maria Mattia, Federico Nemmi, Patrice Péran, Xavier Franceries, and Marie Véronique le Lann. Visual interpretation of cnn decision-making process using simulated brain mri. In *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 515–520. IEEE, 2021.

[18] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[19] Jiachen Yang, Chenguang Wang, Bin Jiang, Houbing Song, and Qinggang Meng. Visual perception enabled industry intelligence: state of the art, challenges and prospects. *IEEE Transactions on Industrial Informatics*, 17(3):2204–2219, 2020.

[20] Chuanfei Hu and Yongxiong Wang. An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images. *IEEE Transactions on Industrial Electronics*, 67(12):10922–10930, 2020.
### Hybrid CNN-Interpreter

#### Figure 9: Summary of filter importance analysis for AlexNet model

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.0130   | 0.0123   | 0.0082   | 0.0068   | 0.0040   | -0.0028  | -0.0090  | -0.0098  | -0.0109  | -0.0111  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.0552   | 0.0332   | 0.0225   | 0.0102   | 0.0003   | -0.0003  | -0.0264  | -0.0265  | -0.0292  | -0.0459  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.0564   | 0.0507   | 0.0422   | 0.0405   | 0.0355   | -0.0002  | -0.0233  | -0.0561  | -0.0581  | -0.0826  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.053    | 0.0518   | 0.0338   | 0.0348   | 0.0003   | -0.0283  | -0.0316  | -0.0422  | -0.0554  | -0.0909  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.1288   | 0.0813   | 0.0775   | 0.0756   | 0.0598   | -0.0698  | -0.0722  | -0.0851  | -0.1045  | -0.1166  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.0605   | 0.0542   | 0.0375   | 0.0196   | 0.0161   | -0.0252  | -0.0288  | -0.0531  | -0.0546  | -0.0597  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.1009   | 0.0710   | 0.0687   | 0.0536   | 0.0457   | -0.0203  | -0.0283  | -0.0366  | -0.0599  | -0.0646  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.0917   | 0.0520   | 0.0358   | 0.0006   | 0.0002   | -0.0454  | -0.0576  | -0.0644  | -0.0688  | -0.0732  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.1248   | 0.0904   | 0.0812   | 0.0778   | 0.0756   | -0.0618  | -0.0744  | -0.0832  | -0.1343  | -0.1439  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.0348   | 0.0277   | 0.0124   | 0.0004   | 0.0002   | -0.0011  | -0.0224  | -0.0278  | -0.0537  | -0.0593  |

| Filter 1 | Filter 2 | Filter 3 | Filter 4 | Filter 5 | Filter 6 | Filter 7 | Filter 8 | Filter 9 | Filter 10 |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 0.1120   | 0.0720   | 0.0383   | 0.0375   | 0.0365   | -0.0511  | -0.0512  | -0.0607  | -0.0693  | -0.0826  |
Figure 10: Summary of filter importance analysis for ResNet model