Visual interpretation Methods for CNNs

Siddhant Agarwal
MIE University of Toronto

Daryl Chua Kee Han
ECE University of Toronto

***Abstract* —** In this paper two visual explainable AI methods for CNNs, Grad-CAM and integrated gradients are evaluated. Both these methods are applied on CNNs trained on the MNIST1D and HMT dataset. Grad-CAM is found to perform better than integrated gradients.

Keywords* —*AI, CNN, Explainable, Grad-CAM, HMT, Integrated Gradients, Machine Learning, MNIST1D, XAI

I. INTRODUCTION

Machine learning has made breakthroughs in the recent years for their applications in Computer vision tasks. However, deep learning models are not easily understood and are usually viewed as black boxes. Thus, there is a lack of trust among the users regarding the solutions obtained from the complex deep learning algorithms. Explainable AI attempts to explain the solutions obtained from these complicated algorithms. In this paper visual explainable AI (XAI) methods are used to investigate the local explanations of convolutional neural network (CNN) models for image classification. By visualizing the features most responsible for a model's decision, visual XAI can help gain trust of the users towards the solutions provided by these deep learning algorithms. The output of the solutions of visual XAI is an explanation heatmap which highlights the image regions that are important for the model's decision making. This paper explores Grad-CAM (CAM based method) and the integrated gradients method (Backpropagation based method) on the 2 datasets, MNIST1D and HMT.

II. EXPLANATION METHODS

Among model explanation/attribution methods, the more model agnostic a method is the more general the method is. The most model agnostic explanation methods only require access to the black box model output given an input. A drawback of model agnostic explanation methods is that they do not exploit meta-information from the model produced from training. This meta-information could be useful in producing explanations for the model. Based on the philosophy of exploiting what is known of the model as much as possible, the least model agnostic explanation methods are chosen. This paper focuses on explaining CNNs and Grad-CAM exploits the fact that the trained convolutional filters are basically feature extractors and explanations could be generated by finding out what features the CNN has learned. Integrated gradients exploits the fact that the CNNs are differentiable models (and the derivatives are Riemann integrable). This section explains Grad-CAM and integrated gradients in detail.

A. Grad-CAM

Research Gap filled

Grad-CAM was designed as an improvement over CAM (class activation mapping). The problem with CAM is that it required the model to have a global average pooling (GAP) layer before the final classification layer. This placed restrictions on the type of CNN models it can explain. Architectural changes or retraining can be done but this change to the model means that CAM does not explain the original model. Gradient-weighted Class Activation Mapping (Grad-CAM) is applicable to a wider variety of CNN models and also specializes to CAM when a model with a GAP penultimate layer is explained [1].

Novelty

As mentioned, CAM requires feature maps to directly precede Softmax layers, making it applicable only to a particular kind of CNN architectures in which global average pooling is performed over convolutional maps just before making a prediction. Such architectures may achieve inferior accuracies when compared with general networks or may simply be inapplicable for other tasks. In Grad-CAM, feature maps are combined using the gradient signal that does not require any modification in the network architecture thus, making it applicable to a wide variety of CNN model-families.

Other methods approach localization by classifying perturbations of the input image. Unlike these, Grad-CAM achieves localization by requiring a single forward and a partial backward pass per image and thus is typically an order of magnitude more efficient [1].

Methodology

The idea behind Grad CAM is that convolutional layers retain spatial information, which is lost in fully connected layers. When one of the trained filters is convolved with an image it produces a feature map which highlights where in the image the feature learned is present. The last convolutional layer contains the most spatial information thus the neurons in the last convolutional layer look for class-specific information in the image. Grad-CAM uses the gradient information from the last convolutional layer to assign importance values to each neuron for a particular decision of interest [1].

To obtain the class-discriminative localization map for any class c, compute the gradient of the score (y^c) before softmax, w.r.t. feature map activations (A^k) of a convolutional layer. These gradients are global average pooled over the width and height dimensions to obtain neuron importance weights (α_k^c) [1]:

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Z is the number of pixels in a feature map [1]. To consider the pixels positively contributing to the score of the class a ReLU is applied to the summation of the product of the neuron importance weights with the feature map activations. This results in a heatmap of the same dimension as convolutional feature maps.

$$L_{Grad\ CAM}^c = ReLU(\sum_k \alpha_k^c A^k)$$

The implementation of Grad CAM used is provided with the supplementary materials for this project in xai_utils.py.

Advantage

Grad CAM lend insights into failure modes of the models by highlighting the pixels that are given higher weightage by

the model (showing that seemingly unreasonable predictions have reasonable explanations) which can be used to solve the issue of misclassification, for e.g. it might happen that the pixel intensity of the true label is less which we get to know from Grad CAM thus, we can identify the reason for misclassification.

For adversarial images Grad CAM visualizations correctly localizes the correct or true class even though the model predicted wrong thus, it is robust to adversarial perturbations/noise.

Dataset bias can be identified from the Grad CAM visualizations as they are more faithful to the underlying model and help achieve model generalization [1]. It is also applicable for CNNs having fully connected layers which is an improvement over CAM.

Apart from image classification Grad CAM is also applicable for tasks such as image captioning etc. Grad CAM is highly class discriminative and interpretable. It is applicable for deeper networks as it uses the gradient information from the last convolutional layer of a CNN to identify the important pixels contributing to the prediction of a particular class, thus avoiding vanishing gradients.

Disadvantage

Grad CAM is model specific that means it is applicable only to the family of CNN architectures. The heatmaps obtained are of the same size or dimension as the last layer convolutional feature maps. For deep networks such as VGG and AlexNet the size of the heatmap obtained would be of 14x14 whereas the original image size is 224x224 [1].

Without the presence of ReLU, localization maps sometimes highlight more than the true class for multimodal inputs and as a result their performance is not good [1].

Localization of objects is imperfect if an image has multiple occurrences of an object from the same class. Localization does not correspond to the entire object [2].

B. Integrated Gradients

Research Gap filled

The pioneers of integrated gradients identified 2 desirable properties that explanation methods should have: sensitivity and implementation invariance. They also showed that many prior commonly used explanation methods do not satisfy both properties. Integrated gradients was suggested as an improvement over these techniques and designed to satisfy those 2 properties (among others) [3].

Novelty

The novelty of integrated gradients is that it was designed from the ground up to satisfy certain useful properties. These properties are argued to be fundamental and hence called axioms of attribution methods. This unique axiomatic approach to creating the method also allows the creators to justify it without evaluating it empirically. In the integrated gradients paper, a list of beneficial properties were listed out, then the integrated gradients method was proven to be the only such method that satisfies all the listed properties [3].

Methodology

Consider, a linear model. A simple explanation heatmap just takes the input and multiply it by the corresponding model weights element-wise. The weights gives one the relative importance of the inputs in contributing to the output. For a non-linear model, take the local linear approximation of the model. The weights of this linear approximation is just the gradient of the model. Multiplying the model gradient with the input element wise is precisely the gradient * input method.

However, the issue arises when the model has a locally flat slope with respect to certain pixels. Gradient * input will lead one to multiply the input with a small number and conclude that the pixels are not salient. However this might not be the case. Consider a CNN for image classification that has been trained. Ideally, the classifier should be robust to intensity perturbations. A image of a panda should still look like a panda even if the intensity values of the important panda features are changed slightly. What is salient are higher level features like the black white pattern or the bear-like appearance. We expect ideal models to be sensitive to changes in these higher order features. Gradient * input captures sensitivity to intensity perturbations locally not sensitivity to high-level feature perturbation.

Sensitivity to high-level features is exploited by Grad-CAM. What integrated gradients does is to make the salient pixels less robust to intensity perturbations. This is done by using a baseline image. This baseline is the black image (matrix with all 0's). One can say the baseline is the image with all features turned off. This baseline model should have a low confidence score for all classes. Now in the matrix vector space as one takes the straight line path from the baseline to the actual image, one gets a continuum of images that start black and get brighter and brighter until one gets the original image. Feeding these images into the model sequentially should produce a differentiable monotonically increasing confidence score of the original predicted class (because the model is assumed differentiable on image matrix space).

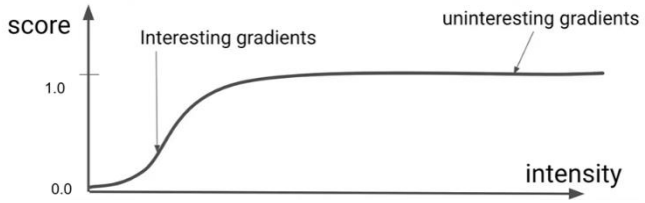


Figure 1: Normalized prediction score of the original predicted class as a function of intensity scaling factor [4].

Suppose the model is robust to intensity perturbations locally on the original image. This is represented by the flat slope at high intensities in figure 1. Because the score has to increase with increasing intensity scale, by the mean value theorem, there has to exist a lower intensity scale where the slope is not flat (or more precisely where the slope has the value of the average slope which is not 0). It is around this region where one expects the salient pixels to be sensitive to intensity changes.

Because of the assumption that the derivatives are integrable, one can use the fundamental theorem of calculus and calculate the mean slope by integrating the gradients along the straight path from the baseline image to the original image. This is the integrated gradients method. The mathematics is formalized in the following paragraph.

Define the model as a scalar-valued multivariate differentiable function of image pixels F. Also the partial derivatives are Riemann integrable. (Image classifiers are

vector-valued but choose only the original predicted class score as the scalar here.) Define **X'** as the **0** matrix and $\mathbf{X_{og}}$ as the original image. The images are flattened into a vector so that the fundamental theorem of calculus for line integrals can be used. Take C as the parametrized straight line curve from the **0** to $\mathbf{X_{og}}$.

$$
\begin{aligned}
F(\boldsymbol{X_{og}}) - F(\mathbf{0}) &= \int^{C} \nabla F(\boldsymbol{X}).\, d\boldsymbol{X} \\
&= \int_0^1 \nabla F(\alpha \boldsymbol{X_{og}}).\frac{d(\alpha \boldsymbol{X_{og}})}{d\alpha} d\alpha \\
&= \int_0^1 \sum_{\boldsymbol{i=0}}^{\boldsymbol{N}} \frac{\partial F(\alpha \boldsymbol{X_{og}})}{\partial X_i} X_{og}^i \, d\alpha \\
&= \sum_{i=0}^{N} X_{og}^i \int_0^1 \frac{\partial F(\alpha \boldsymbol{X_{og}})}{\partial X_i} d\alpha
\end{aligned}
$$

Define the attribution of the i-th pixel as $X_{og}^i \int_0^1 \frac{\partial F(\alpha \boldsymbol{X_{og}})}{\partial X_i} d\alpha$. This is the integrated gradients method (provided the baseline is **0**) [3]. In words, it is the element wise product of the input image and the integrated partial derivatives along the line from baseline to the original image. Note that the integrated partial derivative is equal to the average partial derivative along the path. Hence if one is following the naming convention for gradient * input one can call the integrated gradients the mean gradient * input. Replacing the mean gradient with the gradient at $\alpha = 1$ yields the standard gradient * input method. From the equation above note that the sum of all the attributions of individual pixels is equal to the increase in confidence score from baseline to the final image. This property is known as completeness. Completeness implies the sensitivity property which gradient * input does not have [3].

Because of the assumption that the partial derivatives are Riemann integrable, the integral can be approximated arbitrarily well with Riemann sums.

$$
IntegratedGrads_i \approx \frac{X_{og}^i}{M} \sum_{k=1}^{M} \frac{\partial F\left(\frac{k}{M} \boldsymbol{X_{og}}\right)}{\partial X_i}
$$

The implementation of integrated gradients used is from the tf-explain API on GitHub [5].

Advantage

Compared to Grad-CAM, integrated gradients is relatively model agnostic. The only requirement on the model is that it is differentiable with Riemann integrable derivatives. This includes almost all neural networks (CNNs included).

Integrated gradient is also robust to noise provided that the model is trained to saturation (as in figure 1). As mentioned above, such a model is robust to local intensity perturbations. The model is then 'sensitized' by using a baseline image that has a low prediction score. The mean slope from baseline to the original image will be steeper than that of vanilla gradient. This design allows integrated gradients to be able to handle deep networks and the vanishing gradient problem.

Integrated gradient is also designed to satisfy the following useful axioms: Completeness, sensitivity, implementation invariance, linearity and symmetry preservation [3].

Completeness can be thought of as a 'conservation' of attribution. The sum of all attributions is the change in model output from baseline to original input. This is useful when the numerical output of the model is significant like in time-series predictions [3].

Sensitivity is the property that input variables that change the output should have non-zero attribution score. This direction is implied by completeness [3]. Conversely, input variables that do not affect the output of the model should have zero attribution score.

For all inputs, functionally equivalent models produce the same output. Implementation invariance is the property that attribution scores should be identical for models that are functionally equivalent. Integrated gradients inherits this property from gradients * input [3].

Given one fixed input instance, one can view an explanation method as a map from models to a attribution heatmap. If this map is linear then the explanation method is said to be linear.

Input variables are symmetric with respect to a model if swapping the variable order does not change the model output across the whole input space. An explanation method is symmetry preserving if it gives the same attribution score to input variables that are symmetric with respect to the model.

Disadvantage

Compared to gradient * input, integrated gradients requires about M times the computation time where M is the number of Riemann partitions. This is because each term in the Riemann sum requires a computation of the vanilla gradient at the scaled input.

Integrated gradients is also less model agnostic than methods like LIME that only require black box access to the model output. Here access to the gradients are required.

In the context of CNNs for (1-D or RGB) image classification, integrated gradients is expected to perform poorer than Grad-CAM in explaining and inspecting misclassifications. Integrated gradients detects sensitivity of the model to perturbations in image intensity space and not sensitivity to perturbations in higher-level feature space which Grad-CAM excels at. This is because integrated gradients backpropagates through the whole model to the input while Grad-CAM only backpropagates to the last convolutional layer where the high-level feature detectors are. The completeness property makes integrated gradients more suited for models that perform estimation or prediction rather than classification.

III. MNIST1D RESULTS AND DISCUSSION

MNIST1D is 1D and low memory analogue of the popular MNIST digit classification dataset. Just like MNIST, MNIST1D has 10 classes from 0 to 9. Unlike the original MNIST, MNIST1D is 1D sequence of points obtained by augmenting 1D template for each of the digits by random padding, translating, adding gaussian noise, constant linear analogues to shear in 2D images, finally down sampling the image obtained to 40 data points. A shallow CNN (provided for the project) is trained on 4000 train images, the performance of the model is then evaluated on the test data having 1000 images.

A. *Task 1: Classification Performance*

Overall classification accuracy on the test data is 87.7%.

Table 1 shows the classification accuracy for each class.

TABLE 1: CLASS-WISE ACCURACIES ON MNIST1D TEST SET

Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
98.04 %	80.77 %	88.76 %	94.34 %	84.91 %	81.63 %	91.92 %	86.46 %	91.84 %	78.43 %

Fig. 2 shows the multiclass one-vs-rest ROC and AUC metrics.

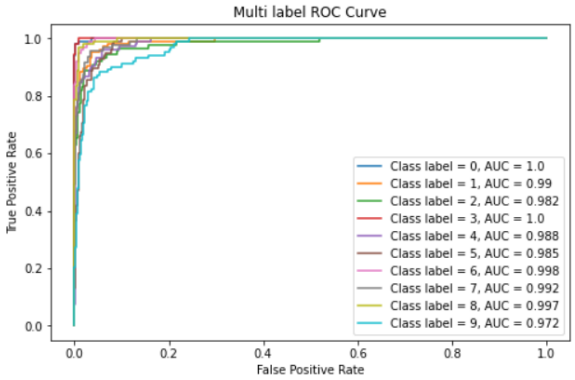


Figure 2: Multiclass one-vs-rest ROC and AUC on MNIST1D test set

Fig. 3 shows the normalized confusion matrix, and the diagonal entries represent the classification accuracy for each class. The normalized confusion matrix is formatted to the nearest 2 decimal places.

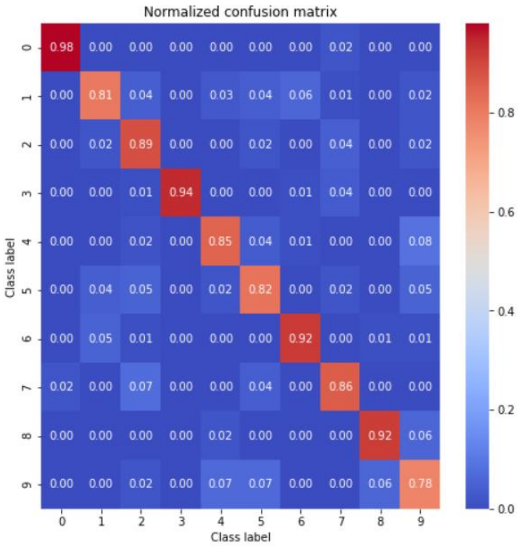


Figure 3: Normalized confusion matrix on MNIST1D test set

Precision, recall and F1 score for each class can be seen in the classification report as shown in Fig. 4.

	precision	recall	f1-score	support
0	0.98	0.98	0.98	102
1	0.88	0.81	0.84	104
2	0.78	0.89	0.83	89
3	1.00	0.94	0.97	106
4	0.87	0.85	0.86	106
5	0.79	0.82	0.80	98
6	0.92	0.92	0.92	99
7	0.86	0.86	0.86	96
8	0.93	0.92	0.92	98
9	0.76	0.78	0.77	102
accuracy			0.88	1000
macro avg	0.88	0.88	0.88	1000
weighted avg	0.88	0.88	0.88	1000

Figure 4: Classification report showing the class precision, recall and F1 scores

Fig. 5 depicts the success case for class 0. Fig. 6 depicts the failure case for class 9. From Table 1, class 9 and class 1 have the lowest classification accuracy of 78.43% and 80.77% respectively. Fig. 7 shows the number of times a particular class is predicted for class 9 and 1. The model is correctly able to highlight the important region for the cases when the data augmentation does not make significant impact on the image. For example, in Figure 5 the image is correctly identified because for the 0 to 10 interval in the input array matches closely with the template and the data augmentation is not significant as to impact the judgement of our model. The model misclassifies in cases when the data augmentation results in an image having features of another class. For example, in Fig. 6, the image for class 9 is wrongly predicted as class 5 because the region in 0 to 10 in the input image closely resembles the template for class 5.

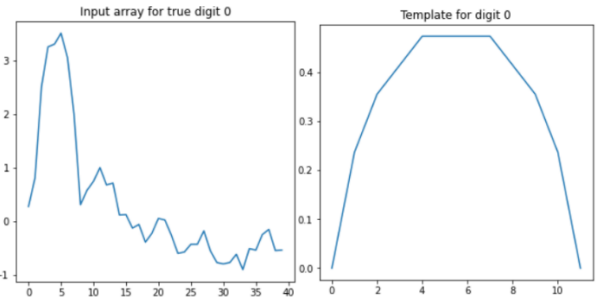


Figure 5: Image correctly predicted as '0'

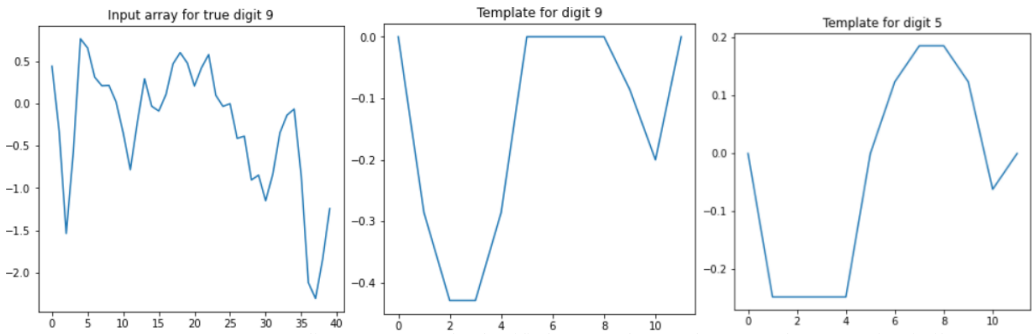
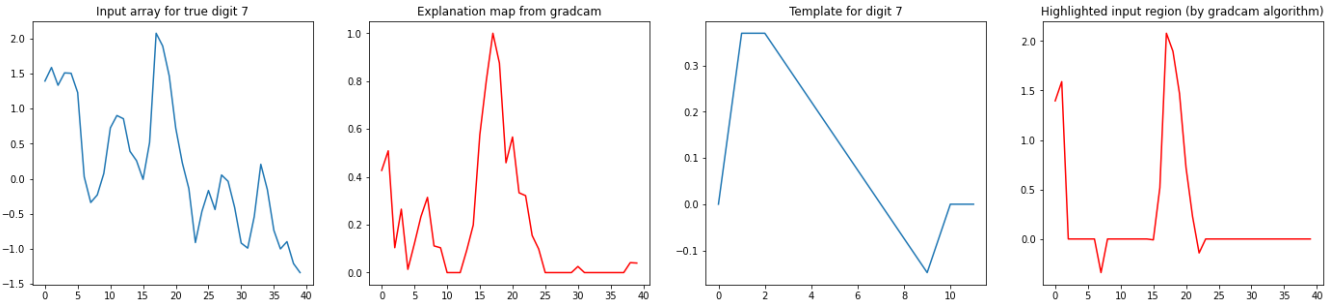


Figure 6: Image corresponding to '9' wrongly classified as '5'. The template '5' and '9' are quite similar.

```
2 times class 2 is predicted when the original class is 9
7 times class 4 is predicted when the original class is 9
7 times class 5 is predicted when the original class is 9
6 times class 8 is predicted when the original class is 9
80 times class 9 is correctly predicted
```

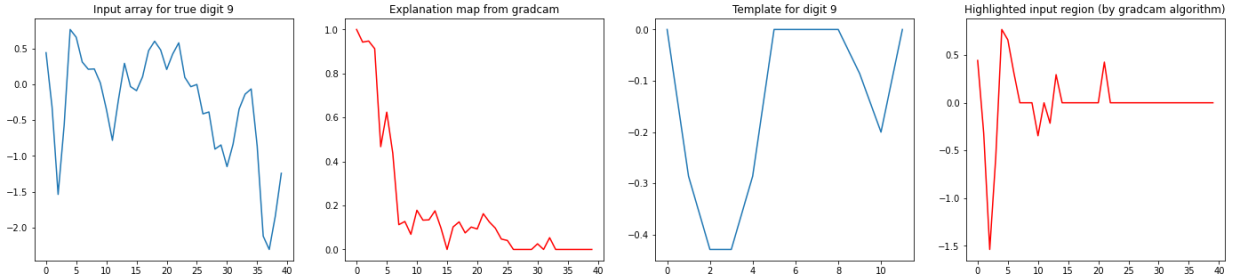
```
84 times class 1 is correctly predicted
4 times class 2 is predicted when the original class is 1
3 times class 4 is predicted when the original class is 1
4 times class 5 is predicted when the original class is 1
6 times class 6 is predicted when the original class is 1
1 times class 7 is predicted when the original class is 1
2 times class 9 is predicted when the original class is 1
```

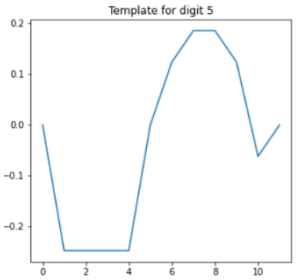
Figure 7: Prediction statistics of class 9 and 1




```
True digit:  7
Predicted digit:  7
Confidence score for the true digit:  0.9615776538848877
Confidence score for the predicted digit:  0.9615776538848877
```

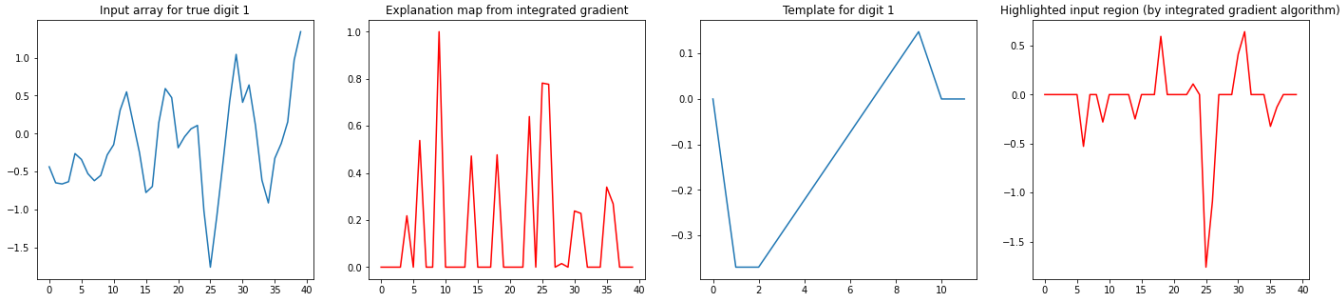
Figure 8: Grad-CAM explanation map for a correct classification of '7'






```
True digit:  9
Predicted digit:  5
Confidence score for the true digit:  0.0005057352827861905
Confidence score for the predicted digit:  0.9722853899002075
```

Figure 9: Grad-CAM explanation map for a '9' misclassified as '5'




```
True digit:  1
Predicted digit:  1
Confidence score for the true digit:  0.726515531539917
Confidence score for the predicted digit:  0.726515531539917
```

Figure 10: Integrated gradients explanation map for a correct classification of '1'

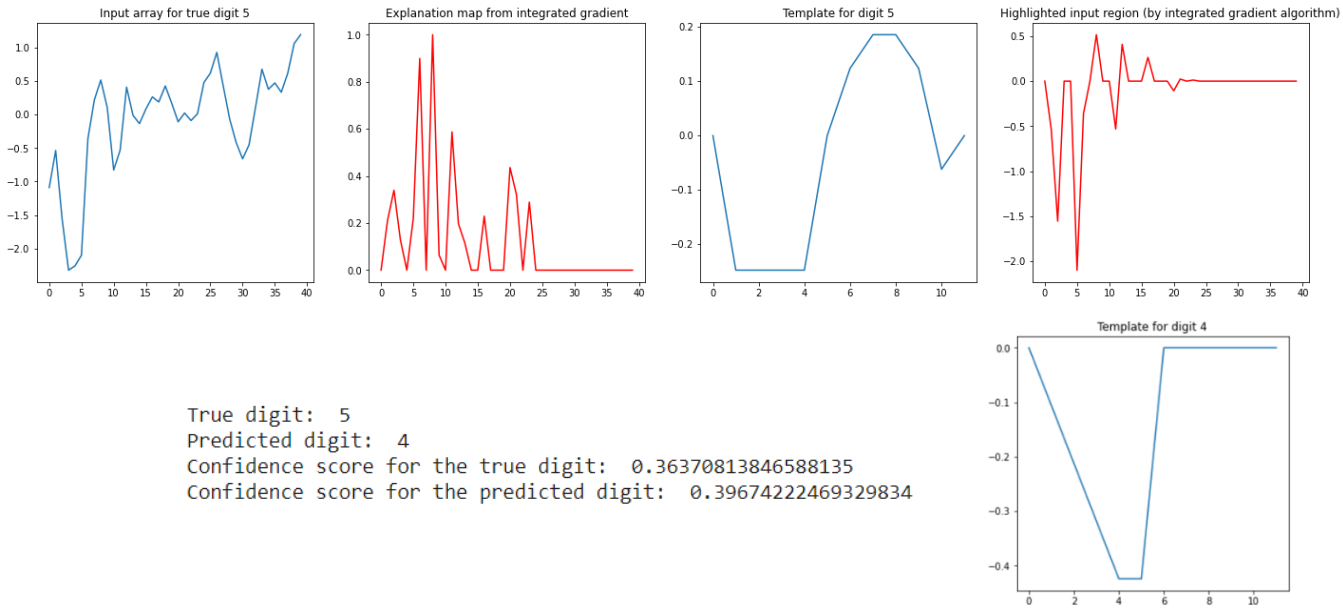


Figure 11: Integrated gradients explanation map for a '5' misclassified as '4'

B. *Task 2: Integrated Gradient and Grad CAM on MNIST1D*

Fig. 8 and 9 show the explanation maps obtained from Grad-CAM for the case when the prediction is correct and incorrect respectively. For the incorrect prediction the predicted template is also provided for comparison. Fig. 10 and 11 show the explanation maps obtained from integrated gradients for the case when the prediction is correct and incorrect respectively. For the incorrect prediction the predicted template is also provided for comparison.

In Fig. 8, the explanation map highlights the region between the range of 15 to 25 as important. In the highlighted region the shape closely resembles the template for the prediction. In Fig. 9 according to the explanation map the region between 0 to 10 is important in determining the prediction. The template for class 9 depicts a flat plateau after a rise but class 5 has a quicker drop after the rise. The highlighted region by Grad-CAM closely resembles that of 5 thus, Grad CAM nicely explains the local behavior of the model at this image.

The figures for integrated gradients specify multiple regions of importance. In Fig. 10 all the peak points of the explanation map are considered to be important and in the image of the top salient points, the region from 25 to 30 closely resembles the template for the prediction. In Fig. 11 all the peak points of the explanation map are considered to be important and in the image of the top salient points, the region between 0 to 5 resembles the template for class 4 more than that for class 5. Thus, integrated gradient nicely explains the local behavior of the model but its results are harder to interpret when compared with Grad CAM as the heatmaps are noisier.

C. *MNIST1D Task 4: Evaluating drop and increase rates*

The intuition behind drop rate is that when unimportant features are removed from the input, the model's confidence score should not drop considerably. Lower drop rate is preferred.

The intuition behind increase rate is that when unimportant features are removed from the input, the model's confidence score might even increase. Higher increase rate is preferred.

For MNIST1D data only the top 30% of salient pixels from the input are considered when calculating for the drop and increase rate on the entire test set. Table 2 shows the drop and increase rate obtained on MNIST1D from Grad-CAM and integrated gradients.

As seen from Table 2, the drop rate for Grad-CAM is slightly more than integrated gradients while the increase rate is much higher for Grad-CAM than integrated gradients. It can thus, be interpreted that Grad-CAM explanations are better than integrated gradients in assigning importance to features. For Grad-CAM 301 images are shown to have higher confidence score when the misleading features were dropped, whereas for integrated gradients only 173 images showed improvement upon removing the misleading features.

TABLE 2: DROP AND INCREASE RATE ON MNIST1D TEST SET

Attribution Method	Drop Rate	Increase rate
Grad CAM	0.411	0.301
Integrated Gradient	0.409	0.173

Both Grad-CAM and integrated gradients can explain some of the model's predictions and fail to explain some predictions. Fig. 12 gives an example when Grad-CAM is unable to correctly explain the models prediction. Due to the data augmentation it seems like there are 2 instances of class 6 (region between 10 to 30). The model correctly predicted '6' but Grad-CAM incorrectly identifies the hill and not the trough as an important feature.

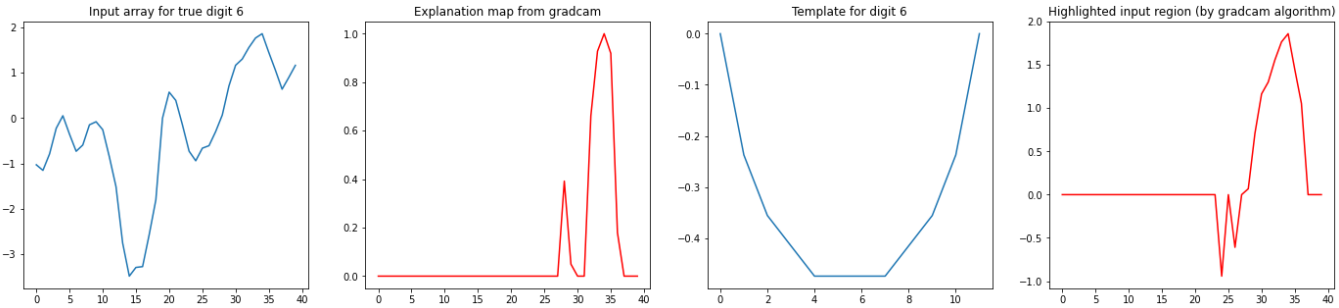


Figure 12: Grad-CAM explanation map for a correctly predicted ‘6’

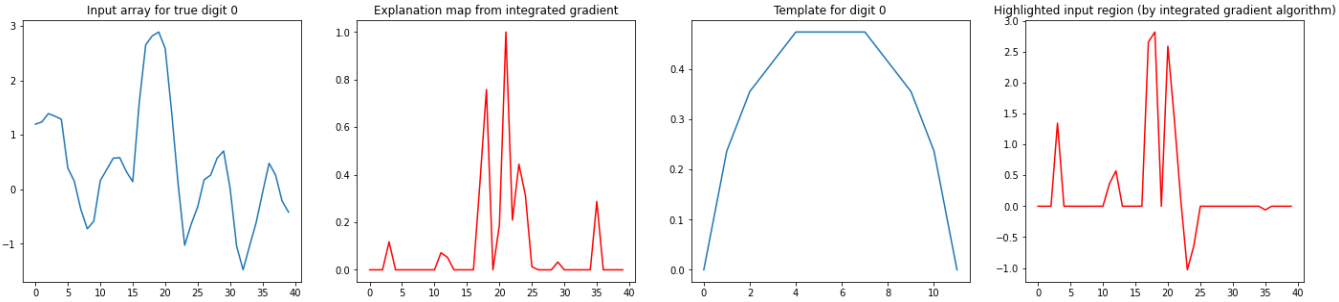


Figure 13: Integrated gradients explanation map for a correctly predicted ‘0’

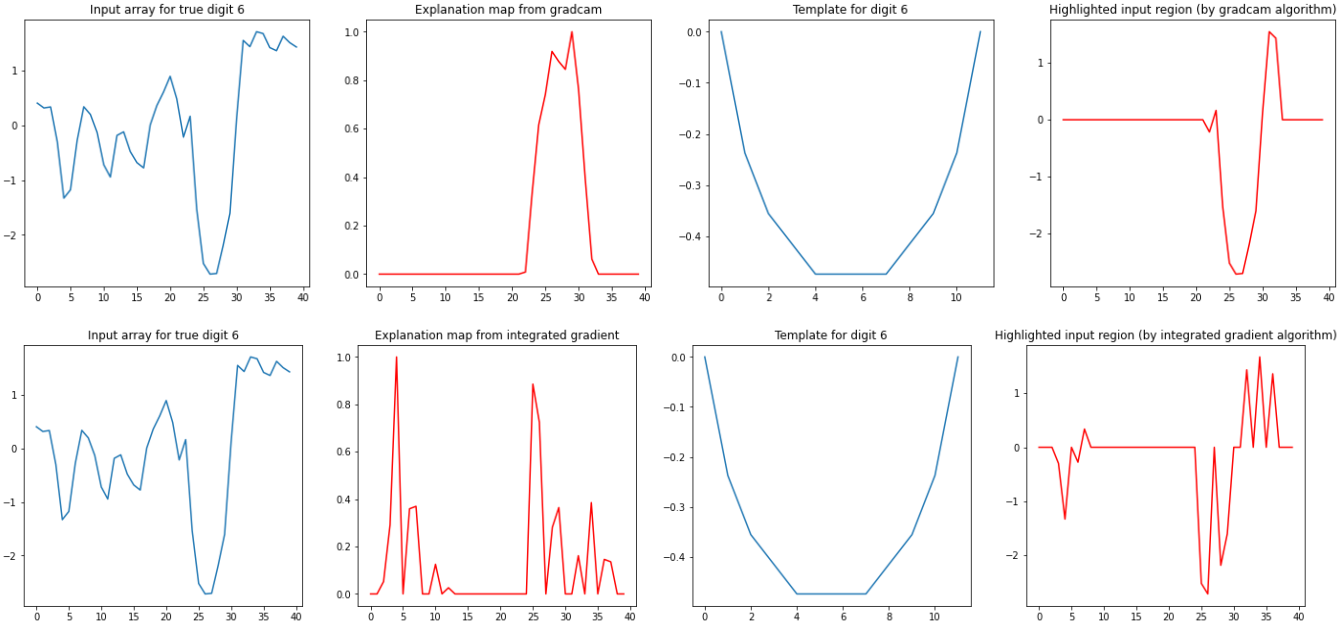
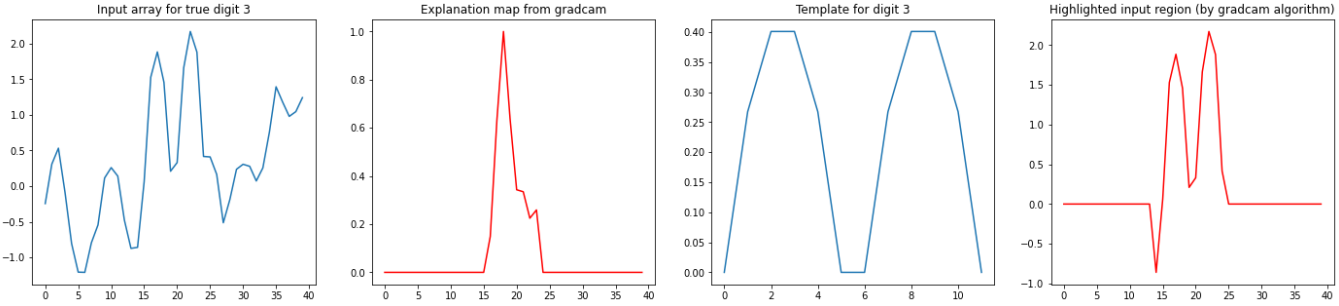


Figure 14: Grad-CAM and integrated gradients explanation map for a correctly predicted ‘6’



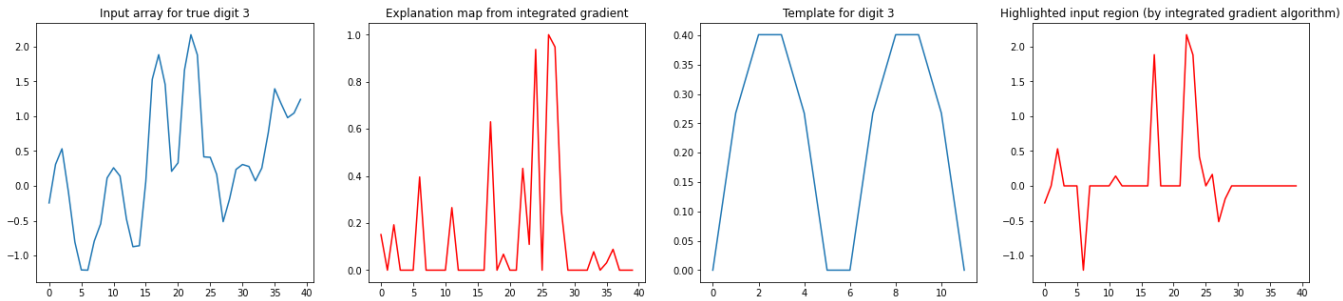


Figure 15: Grad-CAM and integrated gradients explanation map for a correctly predicted '3'

Fig. 13 shows one such case of integrated gradients not explaining the model's prediction clearly. Integrated gradients depict sensitivity to the pixel intensities not high-level features, thus the heatmaps tend to be noisy and the highlighted regions are sparse and scattered. The general area is correctly highlighted but the unimodal shape of the feature is lost and now is highlighted as 2 peaks.

Fig. 14 shows an image that is correctly explained by Grad-Cam but poorly explained by integrated gradients. The explanation is correct from Grad CAM because the data augmentation is not significant, but integrated gradients explanation is too noisy and the top highlighted pixels have split the peaks and troughs into multiple peaks and troughs.

Fig. 15 shows an image that is correctly explained by both integrated gradients and Grad-CAM. It seems like in this case integrated gradients is not too noisy and correctly shows the 2 peaks indicative of a '3'. Grad-CAM also correctly picked the 2 peaks as salient features however it included a dip to the left. This resulted in the high drop rate of 0.868. Perhaps the model has predicted a 1 now. Hence even if the explanation is correct, the drop rate might not be indicative of this.

To summarize, Grad-CAM works better than integrated gradients on the MNIST1D dataset in general. Grad-CAM seems to fail when a class appears multiple times in an image. When Grad-CAM succeeds in explaining an image prediction, integrated gradients sometimes fair worse because it generates a noisy version of the heatmap. The mask generated from the top salient pixels of this noisy heatmap can split the features up into multiple peaks/troughs.

IV. HMT RESULTS AND DISCUSSION

The HMT dataset consists of (histological) microscopic images of human colorectal cancer tissue. There are 5000 (150 × 150 24-bit RGB) images total equally split into these 8 classes: (0) tumor epithelium, (1) simple stroma, (2) complex stroma, (3) immune cell conglomerates, (4) debris and mucus, (5) mucosal glands, (6) adipose tissue, (7) background. 496 out of the 5000 images are partitioned into the test set, providing a train-test ratio of about 9:1. 496 was chosen instead of 500 as it is divisible by 8 classes yielding 62 images per class. The VGG-7 model was trained on this dataset and is to be evaluated and explained.

A. Task 3.1: VGG-7 classification performance

The overall accuracy on the HMT test dataset is 83.47%. The individual class-wise accuracies are shown in table 3 below. Fig. 16 shows the multiclass one-vs-rest ROC and AUC metrics for each class. Fig. 17 shows the normalized confusion matrix. Fig. 18 shows the precision, recall and F1-scores for each class.

TABLE 3: CLASS ACCURACIES OF VGG-7 MODEL ON HMT TEST SET

Tumor	Stroma	Complex	Lympho	Debris	Mucosa	Adipose	Empty
0.85	0.76	0.77	0.85	0.66	0.82	0.95	1.00

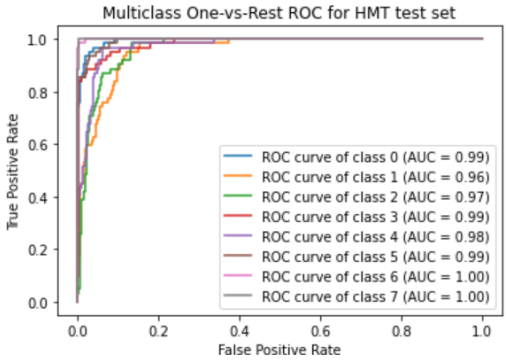


Figure 16: Multiclass one-vs-rest ROC and AUC on HMT test set

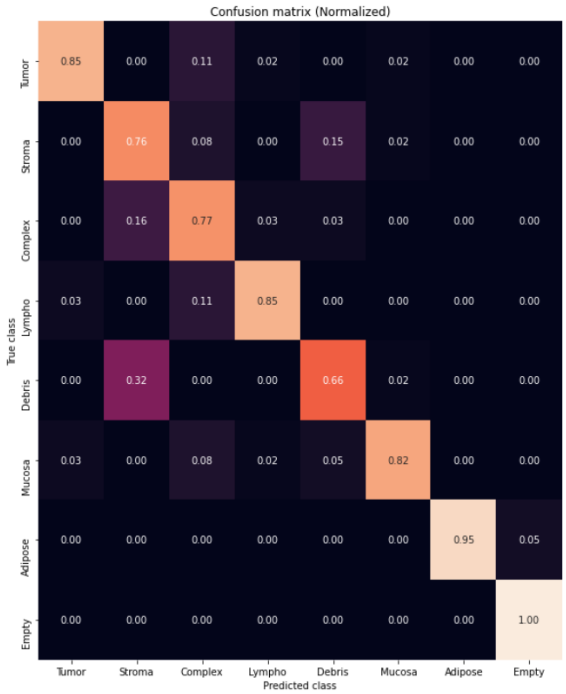


Figure 17: Normalized confusion matrix on HMT test set. The rows have been normalized to sum to 1. Note that the diagonal shows the class-wise accuracies.

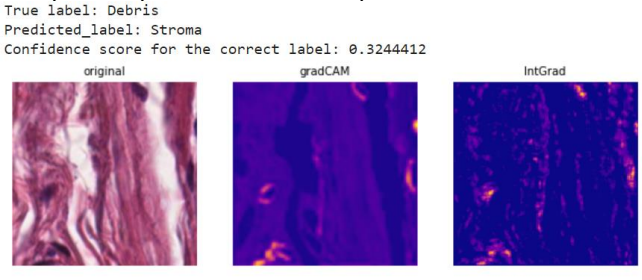
	precision	recall	f1-score
0	0.93	0.85	0.89
1	0.61	0.76	0.68
2	0.67	0.77	0.72
3	0.93	0.85	0.89
4	0.75	0.66	0.70
5	0.94	0.82	0.88
6	1.00	0.95	0.98
7	0.95	1.00	0.98

Figure 18: Precision, recall and F1-Score on the HMT test set

From the confusion matrix in Fig. 17, note that the class simple stroma and class debris and mucus are mistaken for each other quite often. The classes simple stroma and complex stroma are also mistaken for each other but this occurs less often. Finally note that many classes are predicted as complex stroma a significant portion of times, resulting in that column having many non-zero entries. Examples of these cases of misclassification will be investigated in section IV-B below. More precisely, cases that correspond to a value of more than or equal to 0.08 in the confusion matrix are shown.

B. *Task 3.2: Grad-CAM and Integrated Gradients on VGG-7 and HMT test set*

In this section, the explanation methods will be employed on the VGG-7 model and the HMT test set. Note that for a RGB image, the input is effectively a 3D tensor. Hence the heatmaps will be 3D tensors (RGB images) as well. Since one views an image as a 2D matrix of pixel values it is convenient to generate a grayscale attribution maps for each pixel. For grad-CAM this RGB to grayscale process is done by taking the maximum among the 3 channels. For integrated gradients there is a mistake in the implementation by the tf-explain API [5]. RGB to grayscale is done by the summing the absolute values (L1 norm) of the 3 channel values. This method ensures a positive grayscale map but now pixels that contribute negatively to the confidence score are now regarded as salient. The correct way to convert the RGB map to grayscale should be to sum along the channel and then pass it through a ReLU. Note that this grayscale heatmap violates the completeness property. However, since we are just interested in the correct prediction class and not the numerical value of the prediction score, the completeness property is not so relevant. Unfortunately this incorrect RGB to grayscale conversion was discovered quite late into the project and the tf-explain implementation was kept in the interest of time.



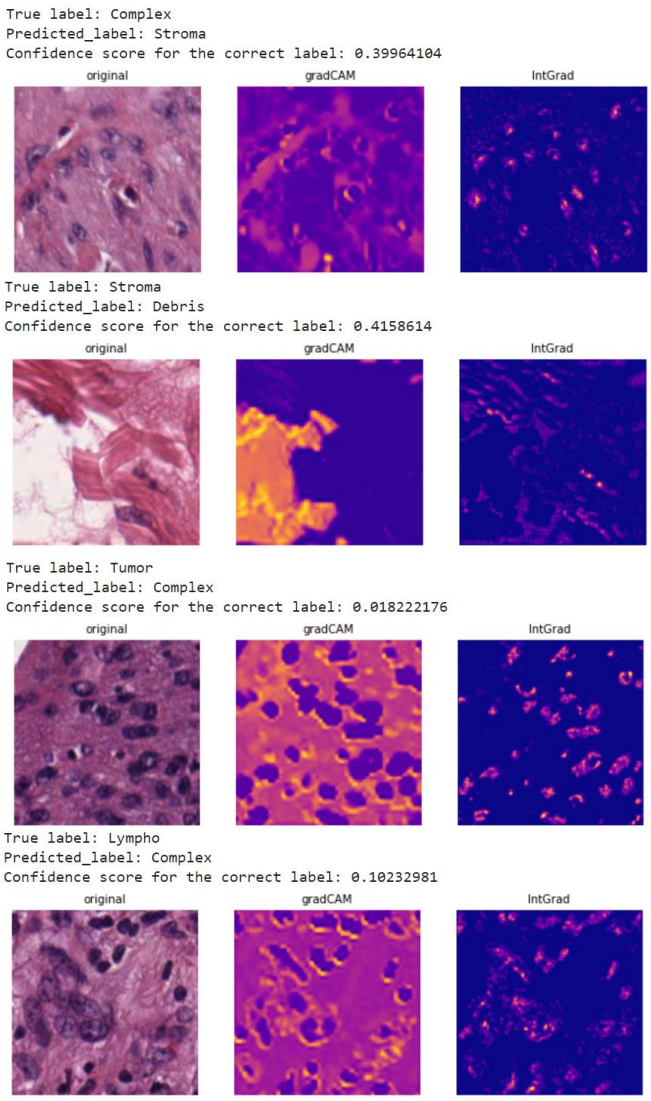
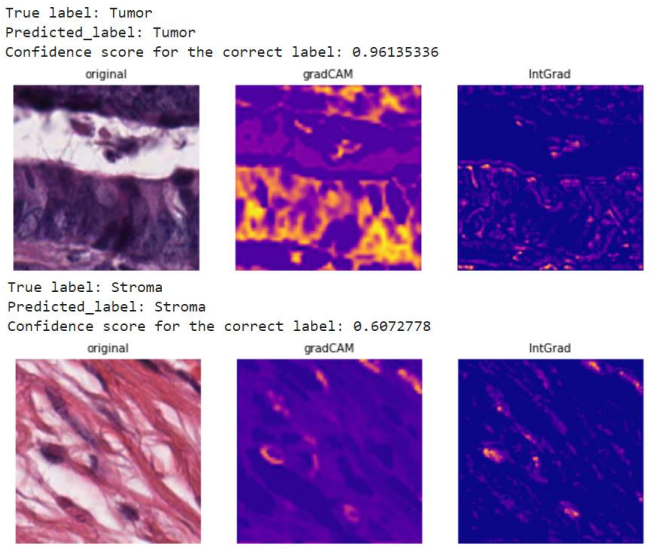


Figure 19: Grad-CAM and integrated gradients for cases corresponding to the top 5 misclassified categories according to the confusion matrix



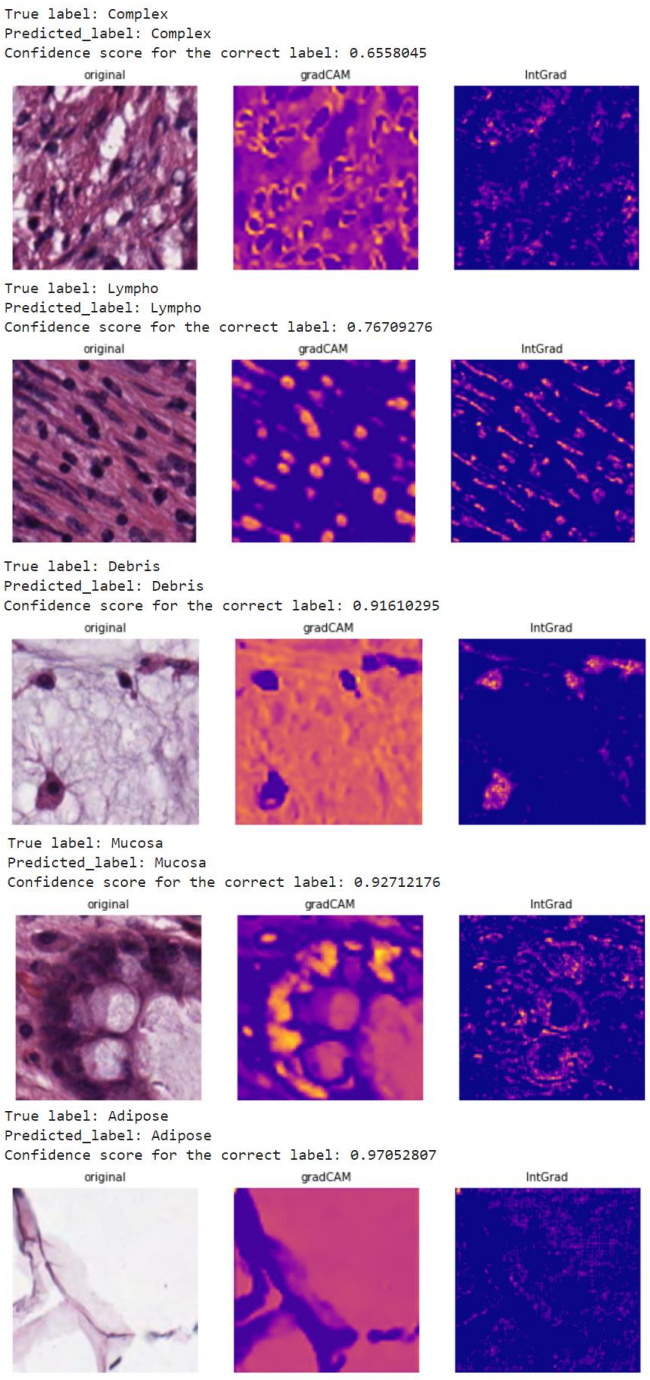


Figure:

Figure 20: Grad-CAM and integrated gradients for cases corresponding to correct predictions for each class

C. HMT Task 4: Evaluating drop and increase rates

For the HMT dataset the top 90% of salient points are retained and the rest are set to 0 to compute the new prediction scores. The decrease in prediction score normalized by the original score is the drop%. The drop% is taken to be 0 if it is orginally negative (the value is ReLUed). The average drop% on the whole dataset is the drop rate. The proportion of images where the drop% is negative is the increase rate.

TABLE 4: DROP AND INCREASE RATE ON HMT TEST DATASET

Attribution Method	Drop Rate	Increase rate
Grad CAM	0.503	0.286
Integrated Gradient	0.885	0.115

As shown in table 1, Grad-CAM explains the model better overall as it has a lower drop rate and higher increase rate. Moreover, an exhaustive printout of the drop% for every image in the test set shows that at best Grad-CAM performs significantly better than integrated gradients. At best the drop rate difference is 1.99 before ReLU which means using Grad-CAM resulted in a huge increase and using integrated gradients resulted in a huge decrease in prediction score. At worst Grad-CAM only has a slightly worse drop rate than integrated gradients (0.04 higher drop rate). This best and worst case scenario are taken as examples to analyze the pros and cons of both methods.

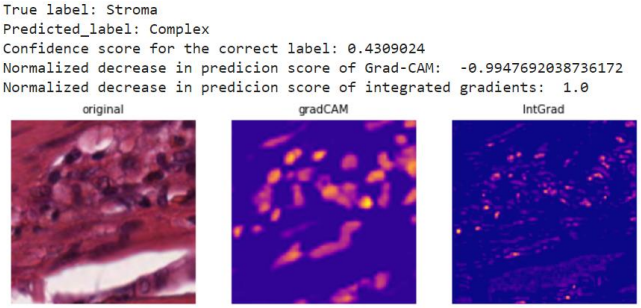


Figure 21: Removing the bottom 10% of Grad-CAM salient points results in almost doubling the prediction score while doing the same for integrated gradients decrease it to 0

In this instance, Grad-CAM performed a lot better than integrated gradients. From Fig. 21 , Grad-CAM has picked up the dark spots as features. Integrated gradients has picked up pixels in the same general vicinity but they are noisy and not recognizable features. This is the advantage of Grad-CAM. Since Grad-CAM does backpropagation to the last convolutional layer, it is detecting sensitivity to changes in the high-level features learned in that layer. Integrated gradients backpropagates all the way to the input. Hence it is mapping out the sensitivity to perturbations in intensity. The former is more important when attempting to explain a CNN for image classification.

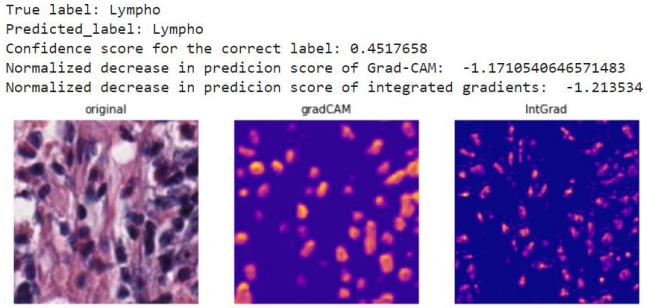


Figure 22: Removing the bottom 10% of Grad-CAM or integrated gradients salient points result in similar increase in prediction score

From Fig. 22, both Grad-CAM and integrated gradients have identified the dark spots as key features. Removing the non-

salient pixels have caused a significant increase in prediction score. Integrated gradients seems to do slightly better. A possible explanation is that the heatmap from Grad-CAM is more blurred compared to integrated gradients. This is a drawback of Grad-CAM. To safe on computational resources, the image loses resolution as it is feedforward through the convolutional layers (of stride > 1) and pooling layers. Since the heatmap is only generated by backpropagation to the last convolutional layer, the resolution at the last convolutional layer determines the resolution of the heatmap which is usually diminished compared to the original image. In this case the resolution is reduced by 4 times across length and 4 times across width.

The next two figures show examples where Grad-CAM and integrated gradients have inverted saliency heatmaps. A possible reason for this is due to the absolute value taken in the integrated gradients implementation instead of the ReLU as mentioned in section IV-B. Most probably, the 'salient' pixels in Fig. 23 as specified by integrated gradients are supposed to have negative attribution scores and would thus lower the prediction scores. Surprisingly the drop rate for Grad-CAM is quite high as well. This implies that both explanations are wrong. Advice from a histologist will be beneficial for the getting the correct explanations and understanding why the explanations are wrong.

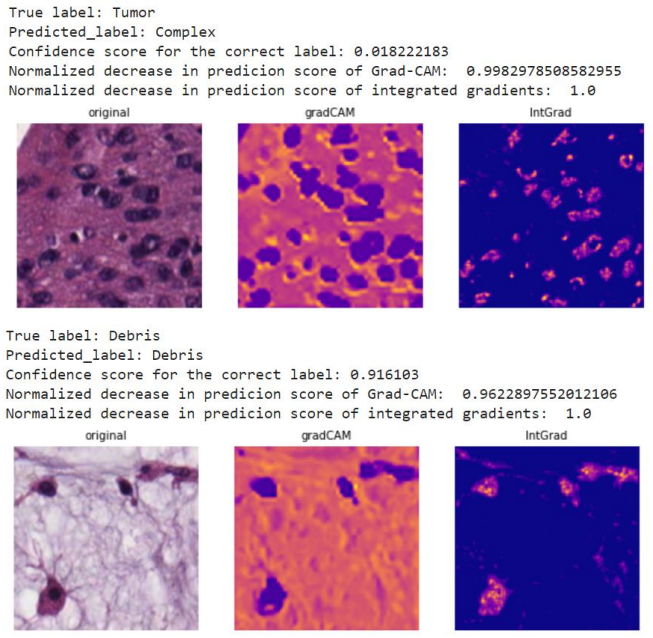


Figure 23: Grad-CAM and integrated gradients showing inverted saliency maps

To summarize, for CNN models where the image resolution reduces through pooling and convolutions, Grad-CAM produces heatmaps of the same resolution as the last feature maps. Integrated gradients does not have this resolution reduction. The drawback of integrated gradients is that it tends to produce more sporadic heatmaps that locate the general vicinity of features but fail to capture the shape of features. This drawback outweighs the resolution decrease of grad-CAM for the VGG-7 model trained on HMT as shown from the drop and increase rate metrics.

CONCLUSION

In conclusion both grad-CAM and integrated gradients have been evaluated as explanation methods for CNN models trained and MNIST1D and HMT. For MNIST1D, grad-CAM cannot explain the model correctly when the image has multiple instances of the same class. Grad-CAM does not suffer from reduced resolution here as the convolution strides are 1 and there is no pooling. Integrated gradients have noisy heatmaps that splits important features up when the top 30% of salient pixels are chosen. For the VGG-7 model on HMT, grad-CAM now suffers from resolution reduction. Integrated gradients suffers from the same problem of noisy and sporadic heatmaps. For both models, the drawbacks of integrated gradients outweigh that of grad-CAM as indicated by the drop and increase rates. This is expected as grad-CAM indicates sensitivity in high-level feature space while integrated gradients indicates sensitivity in image intensity space.

REFERENCES

[1] Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra

[2] Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks, Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, Vineeth Balasubramanian (https://www.researchgate.net/publication/320727679_Grad-CAM_Generalized_Gradient-based_Visual_Explanations_for_Deep_Convolutional_Networks)

[3] Sundararajan, M., Taly, A. and Yan, Q., 2017, July. Axiomatic attribution for deep networks. In *International Conference on Machine Learning* (pp. 3319-3328). PMLR.

[4] *AI Explained Video Series - Learn about Explainable AI and MLOps: What are Integrated Gradients?*. https://www.youtube.com/watch?v=9AaDc35JYiI&ab_channel=FiddlerAI: Fiddler AI, 2020.

[5] "sicara/tf-explain", *GitHub*. [Online]. Available: https://github.com/sicara/tf-explain. [Accessed: 01- Mar- 2021].