Applications of XAI for AI-Generated Image Discrimination

David Blankenship

Introduction

AI image generation technology is continuously improving at a rapid pace. As this technology continues to improve a variety of important social questions around the use of this technology by bad actors to confuse the public with fake images will continue to be a major concern. As such, tools and techniques with the ability to discriminate AI generated images will be necessary and AI techniques can also be part of the solution to this problem. ML classification models can be developed to predict between AI generated and real images, however many of these models are black boxes and understanding why they made these predictions can be critical to improving the performance and gaining the trust of users for the model predictions. This is where explainable AI, or XAI, can be a valuable tool for improving model results and displaying how the models came to their decisions.

In this work, I develop a high-performance classification model and apply several XAI methods to develop an understanding of what features and elements of the image are contributing to the model's prediction. For the machine learning method, I use a modified ResNet18 method. For the XAI methods, I look at Grad-CAM and LIME methods and gain an understanding of how these models explain the results we see and look at patterns between image class and predictions. All work in this paper can be seen in a Jupyter Notebook on my GitHub page at the following link https://github.com/General-Cow/XAI-For-AI-Image-Detection.

Data

The dataset used in this project was sourced from CIFAKE: Real and AI-Generated Synthetic images dataset on Kaggle located at [1]. The dataset contains 60K real images from CIFAR-10 [2] and 60K AI-generated images from [3]. The sources and citations relevant to this dataset can be found in the following references [1-3]. The generated images were generated using the CIFAR-10 images with Stable Diffusion version 1.4. The dataset is presplit into 100K training images and 20K test images with an even split between the fake and real classes. The images are 32x32x3 RGB color images and include 10 categories mentioned in [2] although these categories are not used in this paper.

Methodology

There are two parts to our methodology in this paper: an ML portion for our classification model and an XAI portion for our explanation model. We include the results of the ML in this section as well.

Machine Learning Methodology and Results

The ML model is done in Python primarily using the PyTorch library. The model architecture is based on PyTorch's inbuilt ResNet18 model [4] with a modification to the pooling. In a typical ResNet18 model the pooling would be used to reduce the images to a smaller size, typically 32x32 pixels. Since the images we use are already 32x32 pixels we turn the pooling function into an Identity matrix so that it does nothing to our data. This essentially removes the pooling from the architecture altogether. For our convolutional elements, we use a kernel size of 3, stride of 1, and padding of 1. The initial image starts with a depth of 3 (RGB) and has an initial filter depth of 64. It then follows the ResNet18 model as shown in [4]. There are two output classes: Fake and Real (0 and 1).

For model training, we use criterion and optimizer we use cross entropy loss and Adam with a learning rate of 0.0001. The training is run for 10 epochs. Hyperparameter tuning was not conducted for 2 reasons. First, the ML portion is not the focus of this paper, it is meant to serve as a way to get correct and wrong predictions for both classes. Second, the model had a very impressive performance with a training accuracy of 99% and a test accuracy of 95% on the first run and similar F1-scores for both classes. As this is not the focus, I was happy to use these surprisingly high-performance initial results for the XAI portion of this paper. Table 1 below shows the classification reports for the test data. Figure 1 shows the confusion matrix for the test data.

	Precision	Recall	F1-Score
Fake	0.93	0.96	0.95
Real	0.96	0.93	0.94
Accuracy	0.95		

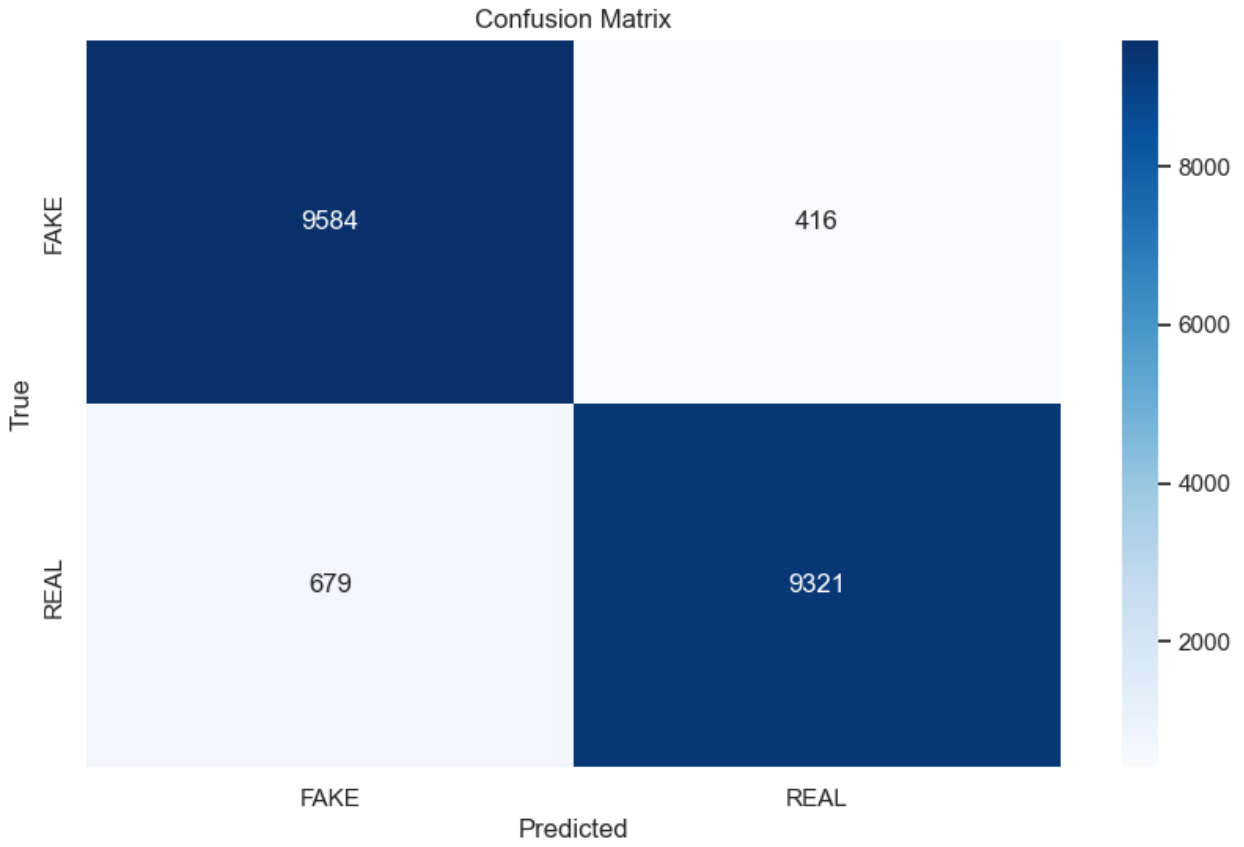


Figure 1: Confusion matrix for test data.

Explainable AI Methodology

For the XAI methods, I elected to focus on two methods, Grad-CAM [5] with the torch-cam implementation [6] and LIME [7]. For both methods, I feed the images into the trained model to predict the class, use the XAI method to create feature maps showing the most important regions for the prediction, and then overlay the feature map onto the image next to the original to allow for a qualitative comparison. The Grad-CAM model has a heatmap overlaid showing which pixels were the most important contributors to the prediction. The LIME method has a simple map overlaid where green is a positive reason for the prediction and red is a negative reason. All code for the XAI methods can be found in my Jupyter notebook at https://github.com/General-Cow/XAI-For-AI-Image-Detection.

Analysis:

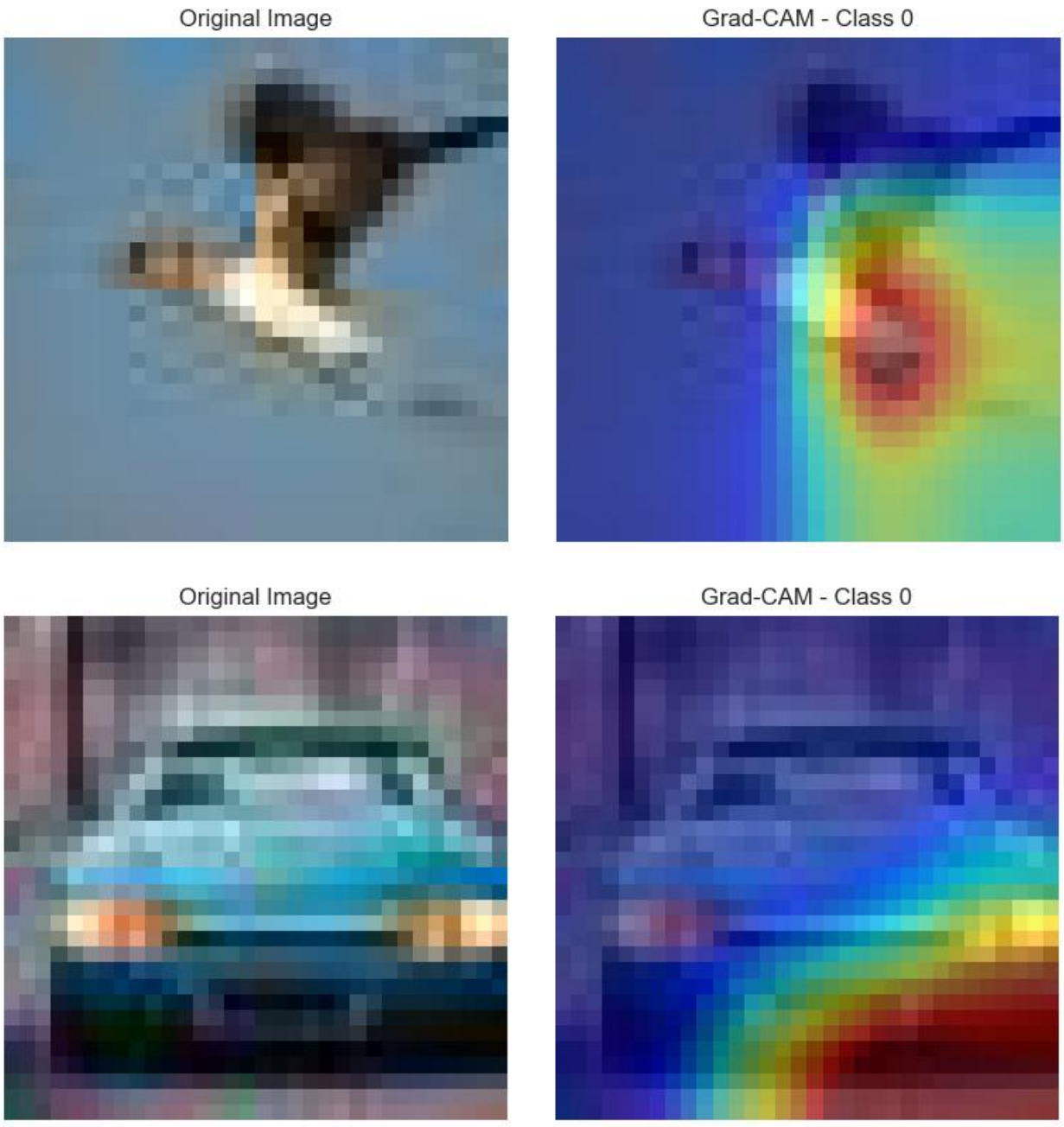
We will begin by looking at the Grad-CAM heatmap images. In our notebook we look at many of the images, but we present a few representative examples here. In figures 2 – 5 we see AI-generated images where figures 2 & 3 are classified correctly and figures 4 & 5

are classified incorrectly. In figures 6 – 9 we see real images where figures 6 & 7 are classified correctly and figures 8 & 9 are classified incorrectly. Qualitatively, it is very difficult to extract any clear patterns across any of the predictions, real or fake, correctly or incorrectly predicted. We see images with one clear spot and multiple spots. We also see heatmaps that focus on the edges and those that don't.

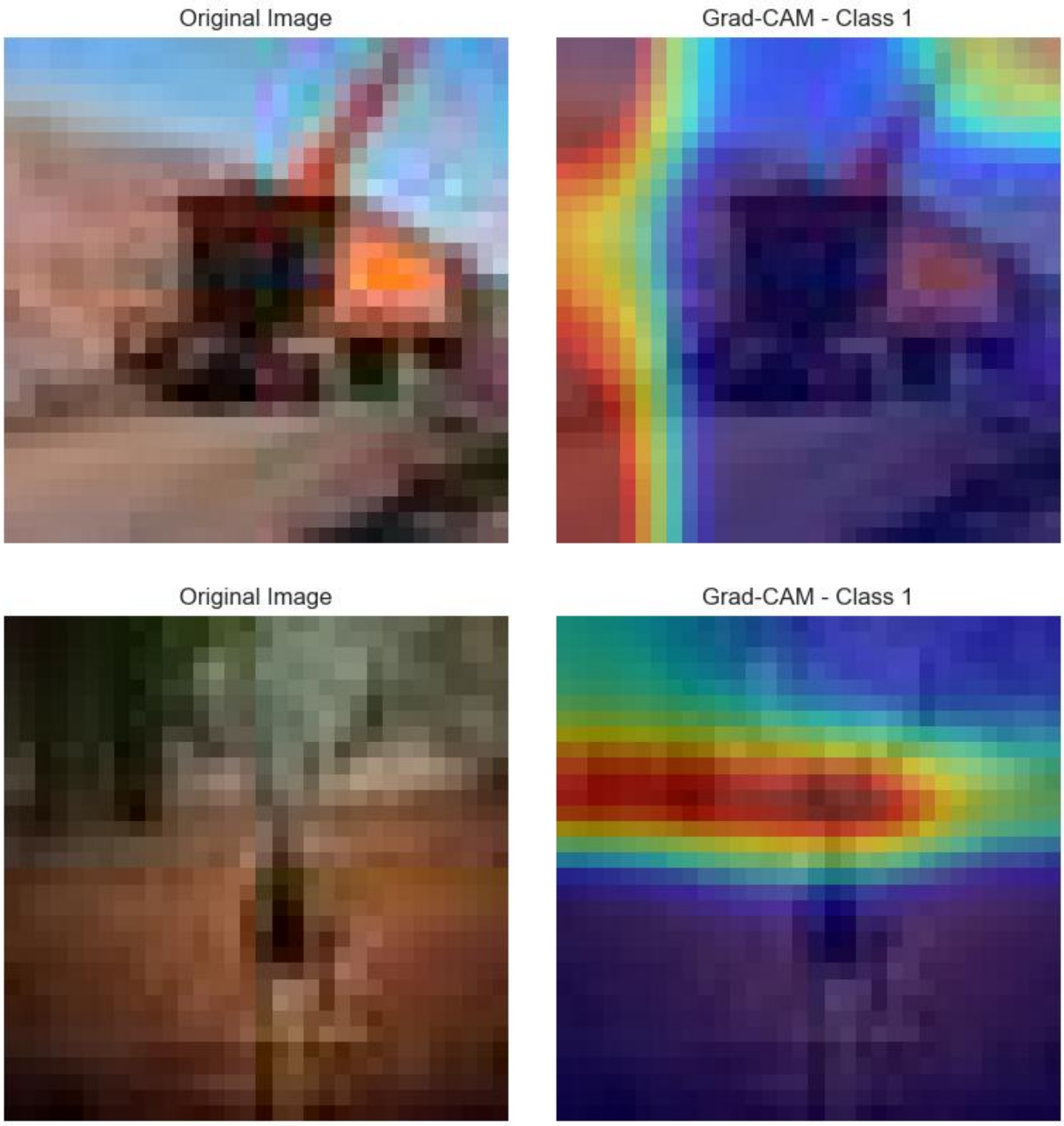
There is also the question of what in particular about the highlighted regions is causing Grad-CAM to declare them regions of interest? Two illustrative examples can be seen in figures 3 and 6. Figure 3 is an AI generated image of a car from the front that was correctly predicted as an AI image. Figure 6 is a real image of a truck at a diagonal correctly predicted as real. They both have the Grad-CAM heatmap showing the lower right corner as the region for why they were predicted correctly. But what about them is being predicted? There is no clear feature that can be pointed to. Even when there is there is no way of knowing how to interpret it. Consider Figure 7 which shows a deer. We see a lighter tuft of fur at the bottom of the image slightly left from the center that seems to correspond with a region of deep importance for why the model was able to predict the image as correct. But we now appear to need an explanation model for our explanation model as there is no way for us to interpret why that region matters. We can go through image after image and, for some, nominally find regions that may correspond to something in the image but there is no way to know what makes that region look real or fake.

I now look at the LIME images and run into an even worse issue. A large majority of the images I looked at are uniformly red, nominally meaning the entire image provides a case against the predicted class. This is the case for all classification scenarios, fake and real, correctly predicted and incorrectly. Figure 10 shows an example of this uniformly negative prediction. It should be noted that Figure 10 is the same original image as Figure 3 for which the Grad-CAM heatmap identified a region of importance. As such we see a disagreement between this LIME model and the Grad-CAM model.

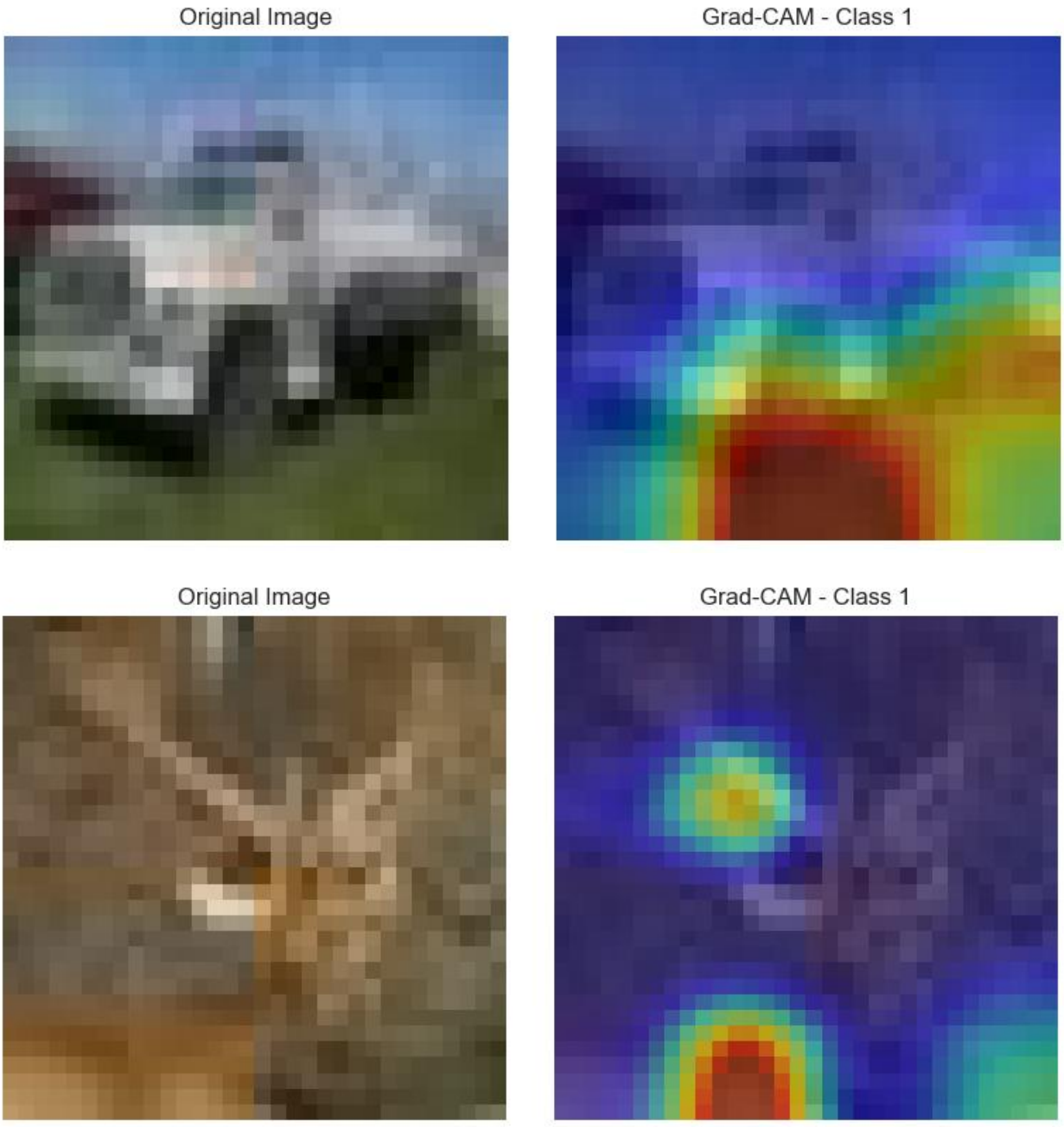
There are some images that show positive regions of prediction such as Figure 11 and 12 which correspond to Figures 2 and 6. However, comparing the regions we see that there isn't a very clear overlap. Figure 11 and 2 clearly don't overlap their regions of interest. Figure 12 and 6 seem vaguely similar, but whether that's a trick of the eye or real is very arguable and potentially grasping at straws. Overall, the LIME method simply does not corroborate even when the prediction is not uniformly negative.



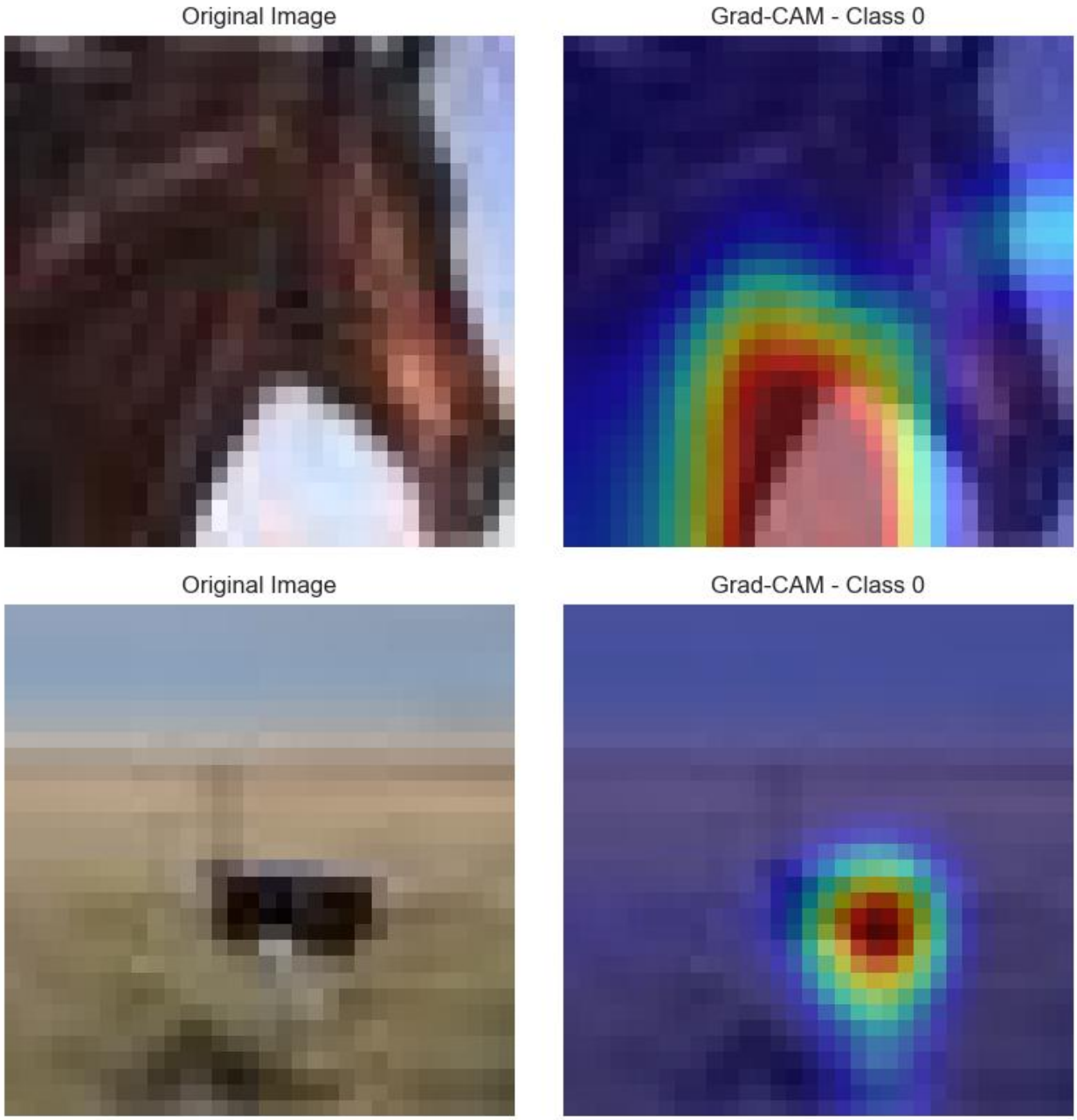
Figures 2 & 3: Fake Images correctly predicted as fake with Grad-CAM Heatmap



Figures 4 & 5: Fake Images incorrectly predicted as real with Grad-CAM Heatmap



Figures 6 & 7:Real Images correctly predicted as real with Grad-CAM Heatmap



Figures 8 & 9: Real Images incorrectly predicted as fake with Grad-CAM Heatmap

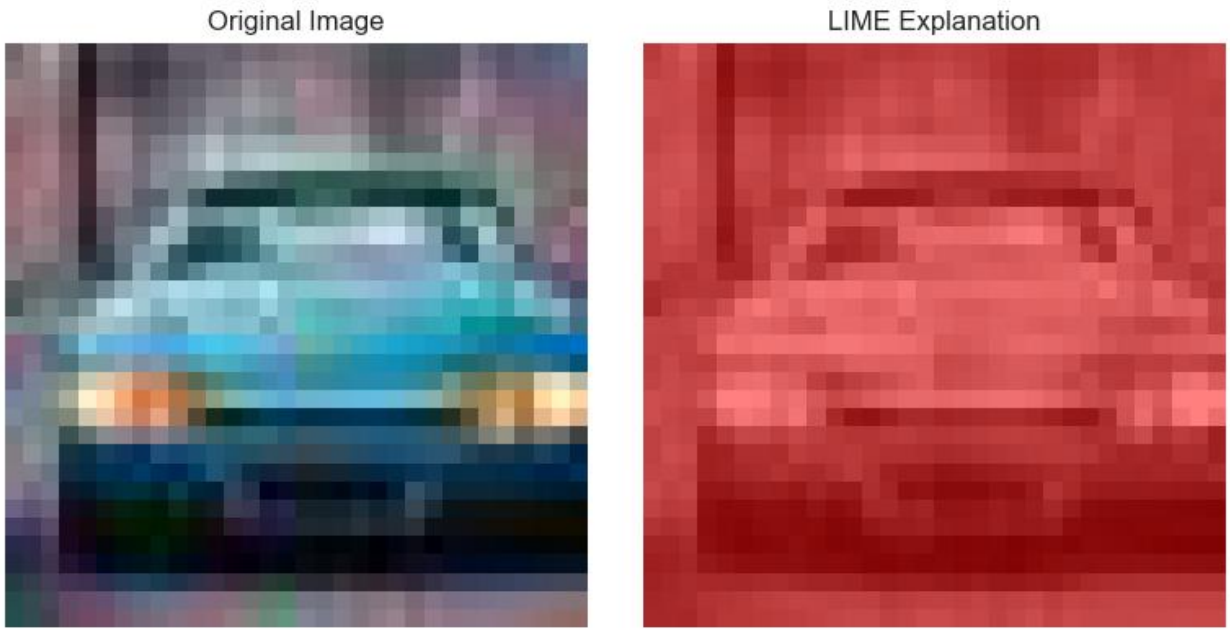


Figure 10: Typical example of LIME output. Corresponds to Figure 3

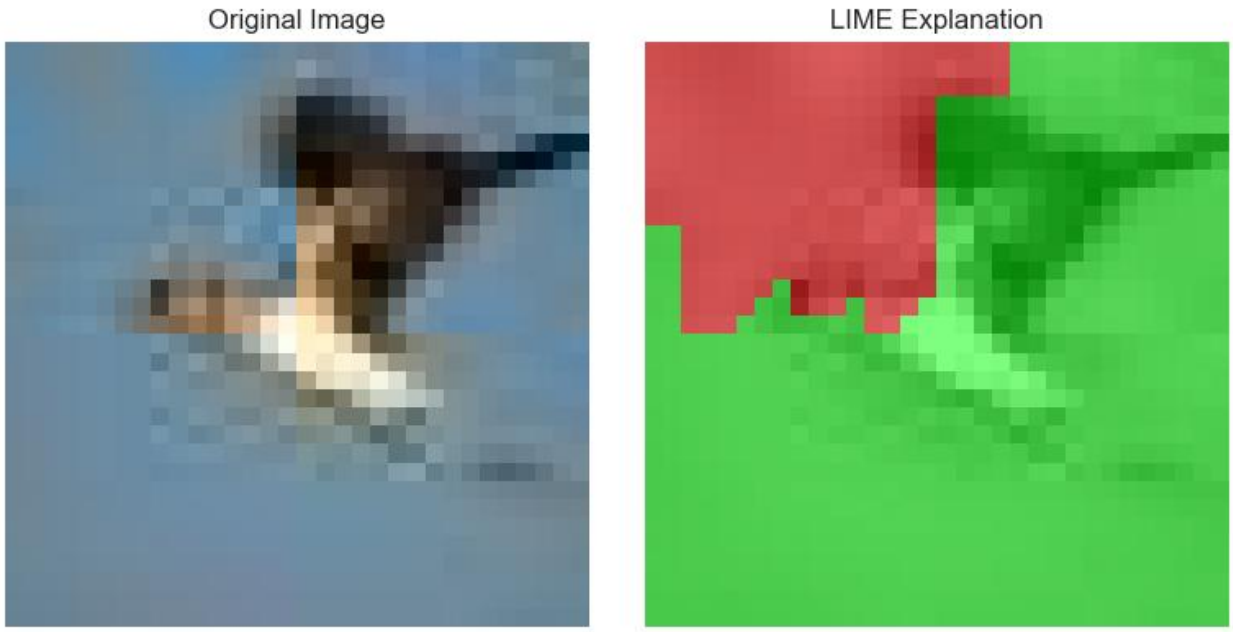


Figure 11: Example of LIME output with green region. Corresponds to Figure 2.

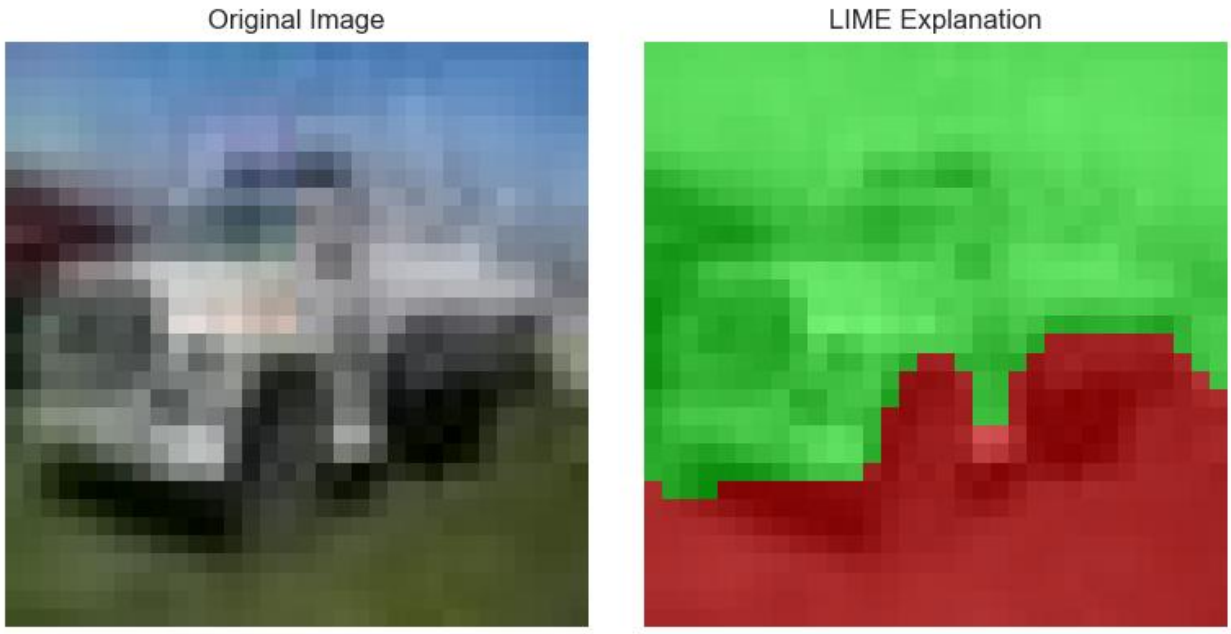


Figure 12: Example of LIME output with green region. Corresponds to Figure 6.

Conclusion:

In this paper we've found a very interesting dynamic between the ML model and the XAI explanations. Our ML model is clearly capable of high performance in distinguishing between real and AI-generated images. However, the explanation we get from the Grad-CAM and LIME feature maps are simply inadequate to explaining what specifically in those regions is causing the prediction. In the original Grad-Cam paper [5] the images were being looked at to explain evidence for classification categories. As such the features are typically very clear, such as a dog picture with the heatmap showing its face. However, with our images attempting to distinguish between real and AI-generated images the specific features the XAI model is using to make the prediction may not even be clearly distinguishable or definable features of the image that can be easily spotted by a human observer.

Consider the case of CNN filters which may be a variety of patterns learned from the images during training. You're going to get abstract patterns that are very difficult to explain why they are relevant. I suspect we have a similar situation here where the specific patterns that distinguish real vs AI-generated images are not easily distinguishable to human identifiable features or mappings. This essentially overrides the explainable portion of XAI. For example, for all I can tell the image may be seeing pixel patterns with a difference of 1 or 2 in value (out of the 256 maximum) that I am completely unable to see. This is just an example, but I have no way of knowing what in the feature regions is the

predictive element and so the explanation is insufficient. We essentially require an explanation for our explanation.

Another major limitation of the methods used in this paper is the issue of scaling this method for evaluating effectiveness. Images are very difficult to handle at a statistical level and this is an issue that goes beyond the methods in this paper. The technique of explaining using feature regions is simply not realistically scalable even within this dataset I used. To give a full account of the data would require going through at least 20,000 images.

There are two major approaches that future work should focus on to improve the XAI methods in this paper. First, there is a need to find explainable methods that are clear to a human. Typical feature map style methods are insufficient when more explanation is required to understand what the feature region is even focusing on. The second is a scalable method for large datasets. Individual images can be handled, but any more than a handful require teams or they can even become outright impossible to go through.

References:

1. Bird, J.J. (2023). CIFAKE: Real and AI-Generated Synthetic Images. Kaggle. https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images/data
2. Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
3. Bird, J.J. and Lotfi, A., 2024. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. IEEE Access.
4. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
5. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).
6. Fernandez, F. (2021). TorchCAM: class activation explorer [Conference paper]. https://s3.amazonaws.com/assets.pytorch.org/ptdd2021/posters/B5.png. https://github.com/frgfm/torch-cam.
7. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).