Project I: Explainable AI for Image Classification using Twin System and Grad-CAM

Aryan Jain
INFO 692: Explainable AI
May 17, 2025

1 Introduction

Explainable Artificial Intelligence (XAI) has become a critical area in understanding the behavior of machine learning models, particularly in sensitive or high-risk applications. In this project, I explored post-hoc XAI techniques applied to a binary image classification task: distinguishing between real and AI-generated (fake) cat images. My goal was not just to train a high-performing model, but also to make its decision-making process transparent.

I applied two complementary XAI methods:

- **Grad-CAM (Gradient-weighted Class Activation Mapping)**: a technique that highlights important regions in an image that contribute most to the model's prediction [1].
- **Twin Case-Based Reasoning (Twin CBR)**: an approach that retrieves similar training examples based on learned embeddings, providing an example-based rationale [4, 5].

2 Dataset

The dataset consisted of 300 cat images:

- 150 real cat images collected from public online datasets
- 150 fake cat images generated using the diffusion model `google/ddpm-cat-256` [2]

Preprocessing:

- All images were resized to 224x224
- Normalized using CIFAR-style mean and std of 0.5

Train/Validation Split:

- 200 images for training (100 real, 100 fake)
- 100 images for validation (50 real, 50 fake)

3 Model

I used a pretrained ResNet-18 model and fine-tuned it for binary classification by modifying the final fully-connected layer to output two classes: real and fake.

Training Configuration

- Optimizer: Adam (learning rate = 1×10^{-4})
- Loss function: CrossEntropyLoss
- Epochs: 10
- Batch size: 32

The final model achieved **91% accuracy** on the validation set.

Evaluation Metrics

To assess model performance, I computed a confusion matrix and classification report:

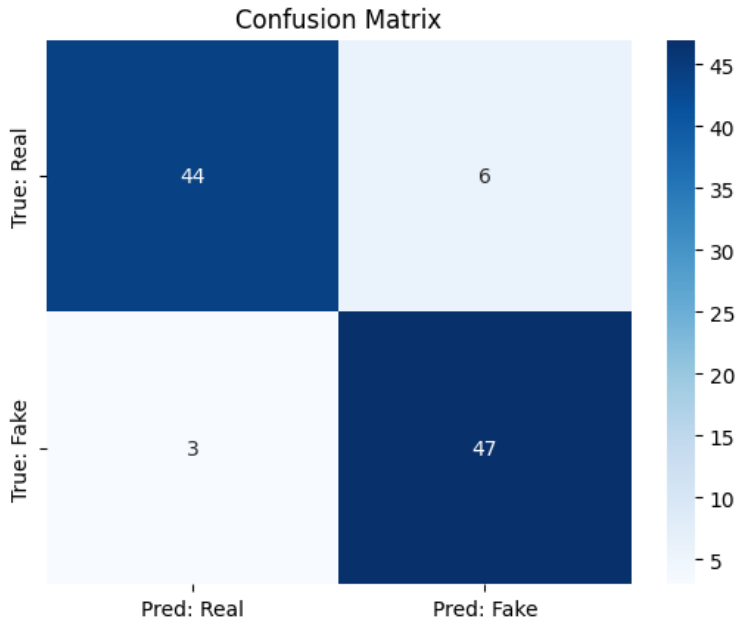


Figure 1: Confusion Matrix and Classification Report

- **Accuracy:** 91%
- **Precision (Real):** 0.94 **Recall (Real):** 0.88
- **Precision (Fake):** 0.89 **Recall (Fake):** 0.94
- **F1-Score (Both Classes):** 0.91

4 Experiments

4.1 Grad-CAM

Library: `torchcam` [3]
Activation maps were extracted from the final convolutional block (`layer4`). The resulting CAM overlays highlight areas of the image contributing most to the prediction.

Visualizations:

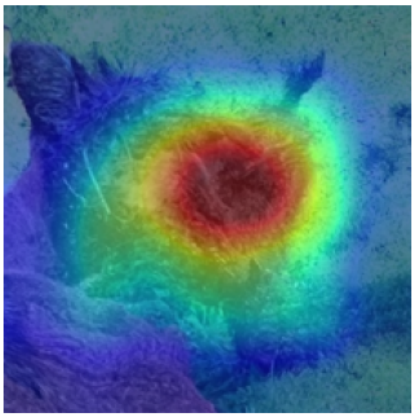
- Original image

- Grad-CAM heatmap overlay

True Label: fake

Original Image

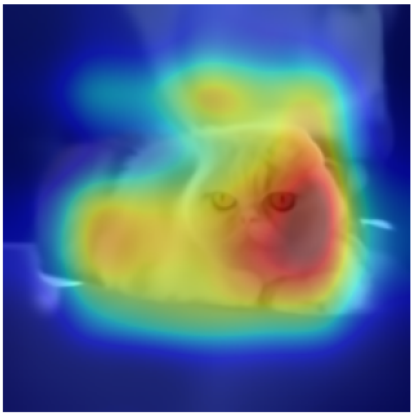
Grad-CAM (Pred: fake)



True Label: fake

Original Image

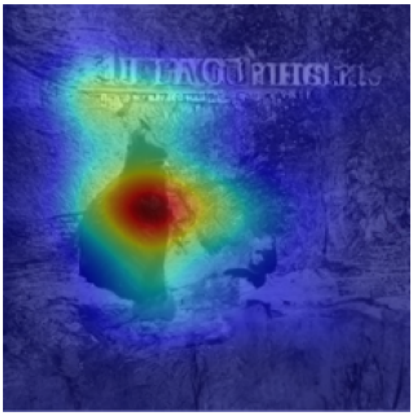
Grad-CAM (Pred: fake)



True Label: fake

Original Image

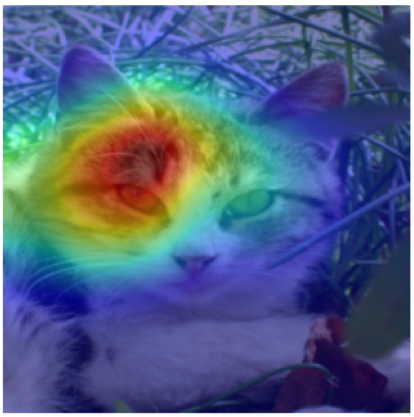
Grad-CAM (Pred: fake)



True Label: real

Original Image

Grad-CAM (Pred: real)



True Label: real

Original Image

Grad-CAM (Pred: real)

True Label: real

Original Image

Grad-CAM (Pred: real)

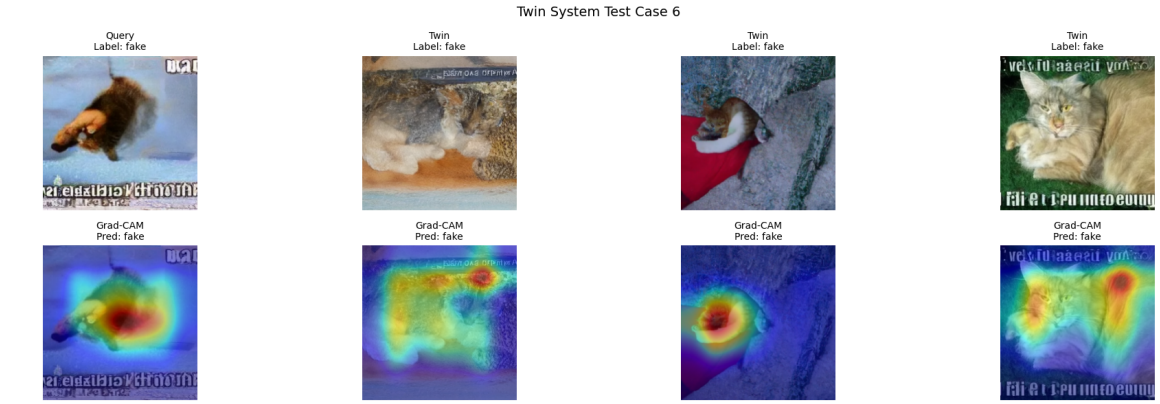
4.2 Twin CBR (Same-Class Neighbors)

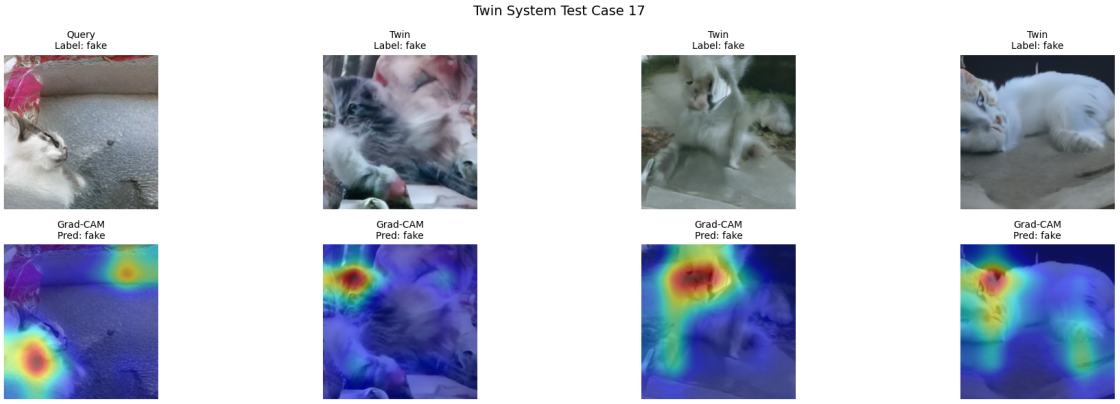
Feature vectors from the penultimate layer (`avgpool`) were used to compute cosine similarity.

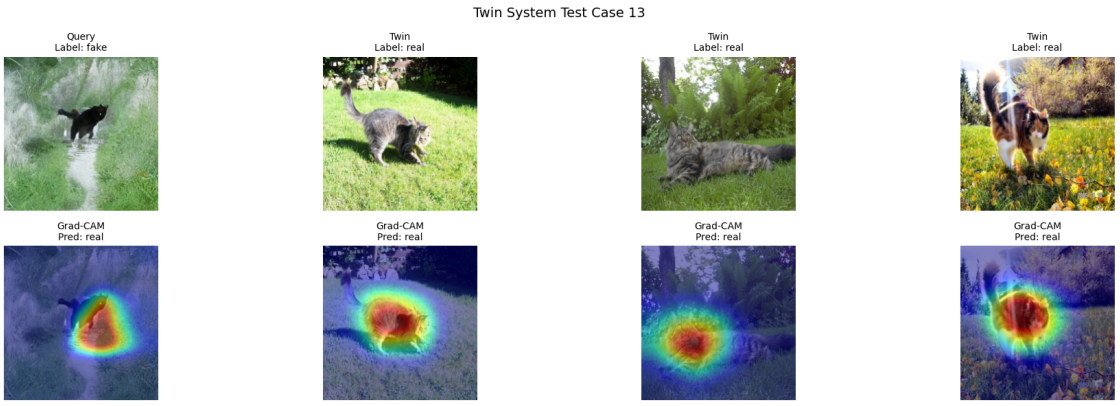
Steps:

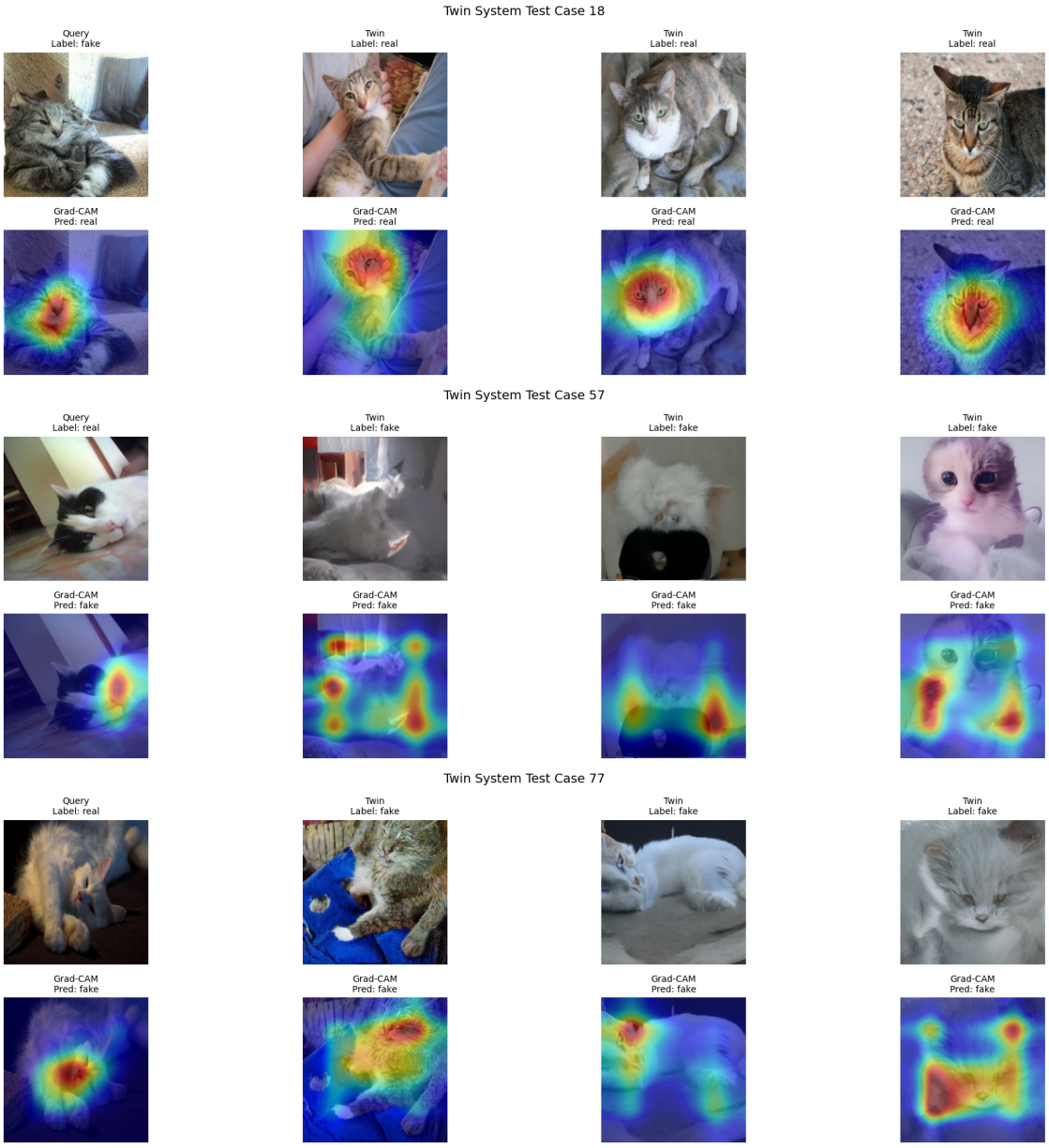
- Extract embeddings for all training and validation images
- Identify $k = 5$ nearest neighbors among training examples **with same predicted class**
- Retrieve neighbors for visual explanation

Justification: The model's decision is supported by visual similarity to known examples.









4.3 Misclassification Analysis

I tracked test indices where the model:

- Predicted "fake" for a real image : Test Cases [13, 18, 22, 34, 40, 44]
- Predicted "real" for a fake image : Test Cases [57, 77, 80]

Twin panels and Grad-CAM overlays were used to interpret these misclassifications.

5 Conclusion

The combination of Grad-CAM and Twin System provided meaningful insights into the model's decision-making:

- Grad-CAM highlights regions that triggered predictions

- Twin explanations give example-based rationales

This dual approach strengthens interpretability and fosters trust in AI image classifiers.

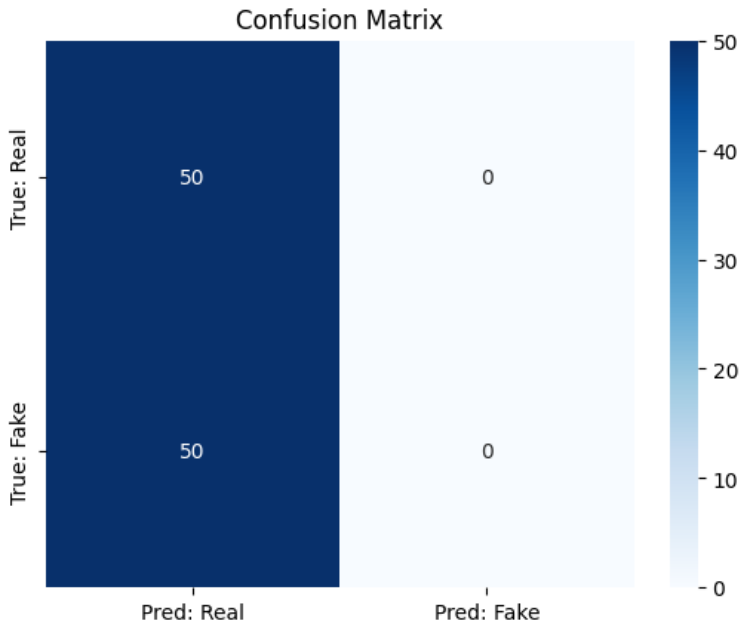
6 Future Work

- Implement counterfactual examples (nearest neighbors from opposing class)
- Improve similarity search using CLIP embeddings
- Train a prototype-based network (ProtoPNet)

ProtoPNet Attempt

I attempted to implement a ProtoPNet model using a ResNet-18 backbone with 10 prototypes per class. The architecture was designed to learn localized representations and classify images based on their similarity to learned prototypes. Although the implementation completed training, the performance was poor:

- **Validation Accuracy:** 50%
- **Class imbalance:** All samples predicted as real



Despite the low accuracy, I was able to visualize the learned prototypes:

The model overfit to the "real" class and failed to activate relevant prototypes for fake images. I plan to revisit this architecture in future work by introducing regularization and prototype pruning.

References

[1] 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.* Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[2] Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising Diffusion Probabilistic Models.* Advances in Neural Information Processing Systems (NeurIPS).

[3] TorchCAM Library, `https://frgfm.github.io/torch-cam/`

[4] Chen, C., Li, O., Tao, C., Barnett, A. J., Su, J., & Rudin, C. (2018). *This Looks Like That: Deep Learning for Interpretable Image Recognition.* arXiv preprint arXiv:1806.10574.

[5] Barnett, A. J., Schwartz, F. R., Tao, C., Chen, C., Ren, Y., Lo, J. Y., & Rudin, C. (2021). *A Case-Based Interpretable Deep Learning Model for Classification of Mass Lesions in Digital Mammography.* Nature Machine Intelligence, 3(12), 1061–1070.