3.3 Image Classification and Processing

Course: The Artificial Intelligence Ecosystem (C202506-01)

Instructor: Chelsea Myers

Student Name: Alvaro Torres

GitHub Repo Link: https://github.com/alvarotorrestx/AI-Image-Processing-Classification

Final Report

Summary:

Part 1: Using the Basic Classifier and Implementing Grad-CAM

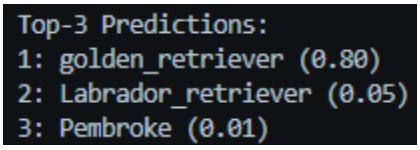
Top 3 Predictions and Confidence Scores

- golden_retriever (0.80)

- Labrador_retriever (0.05)

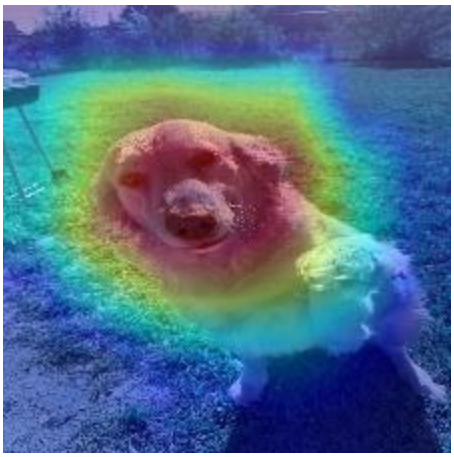
- Pembroke (0.01)

Screenshot:



Heatmap Analysis

Screenshot:



Observation:

The classifier seems to be focusing mostly on my golden retriever's head and face to help identify what the image is. The heatmap shows the hottest areas concentrated around her nose, eyes, ears, and the overall shape of her head.

Part 2: Experimenting with Image Occlusion

Top 3 Predictions and Confidence Scores

Classifying original image: kami.png

- golden_retriever (0.80)
- Labrador_retriever (0.05)
- Pembroke (0.01)

Results for occl_black.jpg:

- spotlight (0.52)
- bubble (0.13)
- stage (0.05)

Results for occl_blur.jpg:

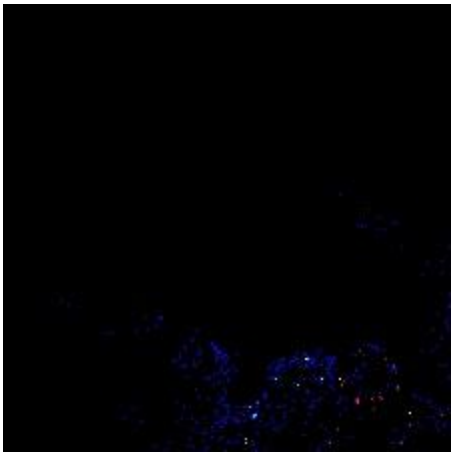
- spotlight (0.53)
- bubble (0.13)
- stage (0.05)

Results for occl_noise.jpg:

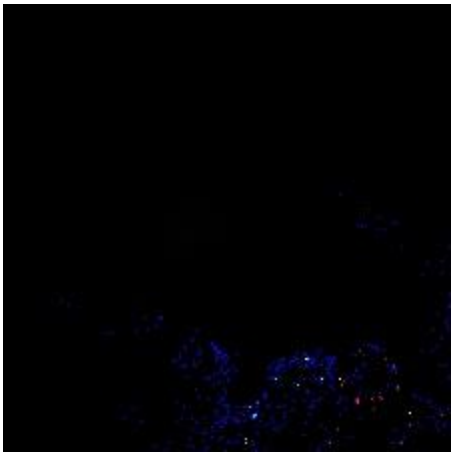
- electric_ray (0.10)
- lampshade (0.09)
- umbrella (0.09)

Screenshot:

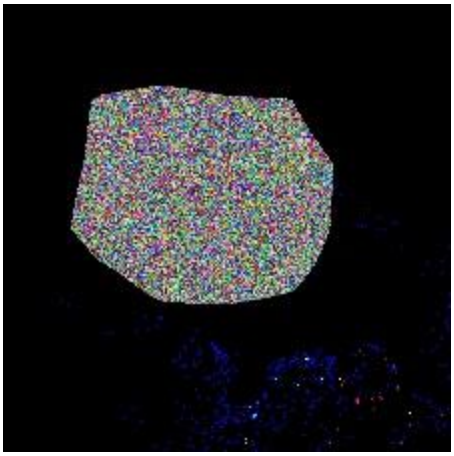
Black Box Occlusion



Gaussian Blur



Noise Injection



Occlusions Analysis

Observation:

After applying three different occlusion methods on the Grad-CAM heatmap, the classifier’s performance dropped SIGNIFICANTLY. The top prediction changed from “Golden Retriever” to unrelated objects like “Spotlight” and “Electric Ray,” with much lower confidence scores.

Did the classifier struggle to classify the occluded images?

Yes, the classifier drasticly struggled once the Grad-CAM was occluded. In the original image, the model correctly predicted golden_retriever with 80% confidence. After applying each occlusion, the predictions came out with completely unrelated objects such as spotlight, electric_ray, and lampshade, with significantly lower confidence scores.

Which occlusion had the greatest impact on performance?

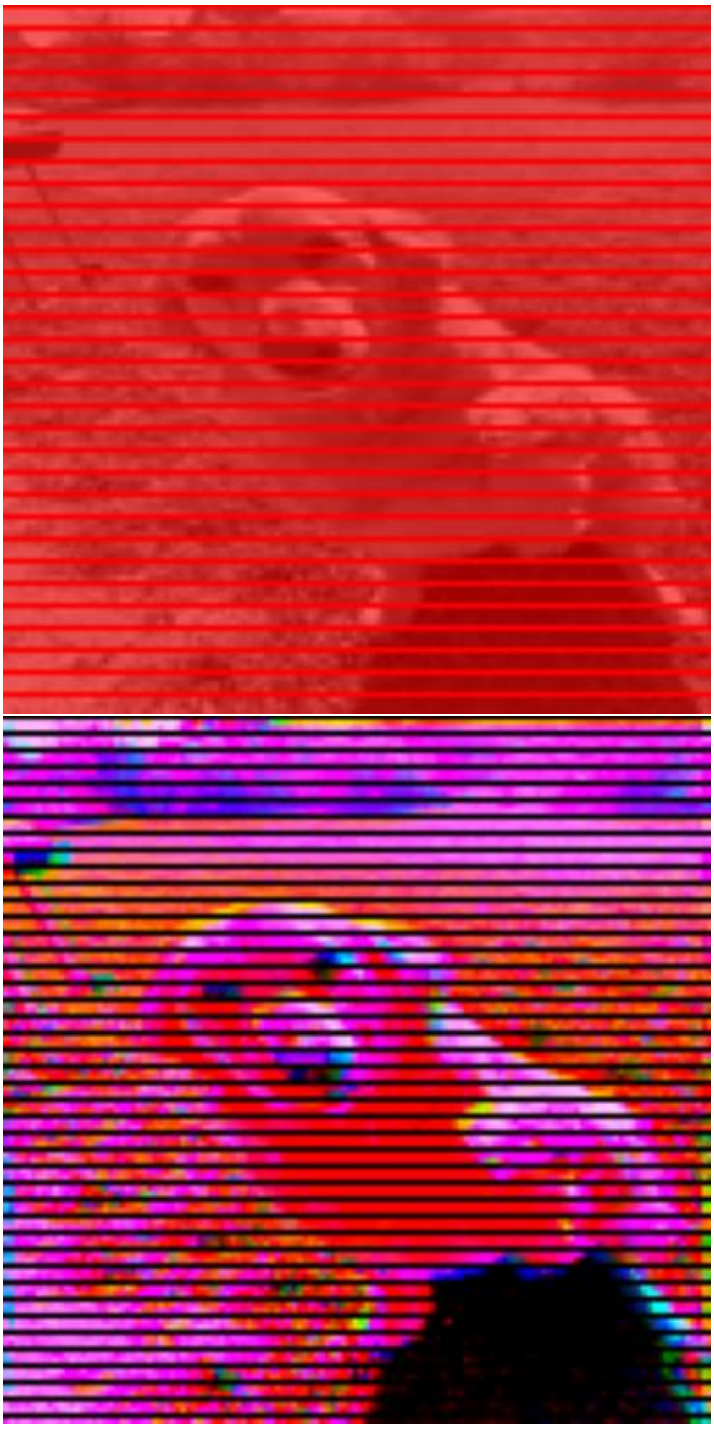
The noise injection had the greatest impact. Not only did the classifier fail to recognize the original object, but the top-3 predictions became unrelated and confidence was extremely low (all around 9–10%). Even though the black box and blur also caused the model to misclassify the image as a "spotlight," they still held slightly more consistency and confidence. The noise disrupted both structure and texture, leading to a complete loss in prediction reliability.

Part 3: Creating and Experimenting with Image Filters

Filter Addition

- Batman Beyond Infrared Vision

- Cyberpunk Glitch



Classifier Behavior & Occlusions:

From the heatmap analysis, it was clear that the classifier heavily relied on specific regions of the image to make accurate predictions. Once the highlighted areas identified by the Grad-CAM were occluded, the model's performance dropped significantly. With the original image, it correctly classified my golden retriever with over 80% confidence. However, once occlusions like blur, black box, or noise were added, the predictions shifted to unrelated objects like "spotlight" or "lampshade," and the confidence levels dropped heavily. Among the three, noise injection had the most drastic impact, essentially corrupting the structural and textural cues that the model needed.

Filters Developed:

I added two custom visual filters for image processing:

1. **Batman Beyond Infrared Vision** – This effect overlays red lines across the image to simulate a stylized infrared look, slightly brightening the image and enhancing contrast. It gives the impression of thermal or "night vision" mode.
2. **Cyberpunk Glitch** – This filter adds RGB channel shifts and magenta-blue tinting to create a digital glitch aesthetic. It combines high contrast and color saturation and evokes a futuristic, almost radioactive vibe.

Working with the AI:

Collaborating with the AI to write and modify Python code was mostly smooth. The AI understood the goals and helped build functional filters quickly and offered useful ideas along the way. Although, it was not perfect! It did take several iterations of exploring before finally deciding to stick with the 2 created. Aside from that, I did get a better understanding of how the code worked and why it was needed. The AI did an amazing job of helping with that.