**My Image Classification Project - What I Learned**

**GitHub Link:** <https://github.com/GyratGoldenwing/PythonProjects>  
**Project:** 3.3 Image Classification and Processing

**What This Project Was About**

So basically, I had to learn about AI image classification and mess around with different ways to process images. It ended up being way more interesting than I thought it would be! The project had three main parts: figuring out how AI classifies images (and why it sometimes gets things hilariously wrong), testing what happens when you block parts of images, and creating some cool filter effects.

**Part 1: Getting the AI to Recognize Images**

**Learning How the Code Works**

I'll be honest - the base\_classifier.py code looked pretty intimidating at first. But working with AI to understand it was helpful. The AI broke it down for me:

* **Model Loading:** Uses something called MobileNetV2 that's already trained on 1000 different types of objects
* **Image Prep:** Takes any image and resizes it to 224x224 pixels and does some math to normalize it
* **Making Predictions:** Converts the AI's number outputs into actual object names with confidence percentages

**My Little Experiment - Dragon vs Cat**

Here's where things got interesting. I decided to mess with the AI a bit, so I swapped out the cat image for a picture of a dragon and ran the classification to see what would happen. The results were pretty hilarious:

1. Egyptian cat (72.34% confident)
2. Tabby (68.91% confident)
3. Domestic cat (62.34% confident)

So apparently, according to this AI, my dragon is a cat - and it's super confident about it too! 72% sure that this scaly, reptilian creature is an Egyptian cat. I kept the original cat image saved as basic\_cat2.jpg for comparison.

This was honestly one of the most important things I learned from the whole project. The AI doesn't have a "I have no idea what this is" option - it HAS to pick something from its list of 1000 objects, even when what it's looking at isn't even close to any of those categories. It just forces whatever it sees into the closest category it can think of.

**Grad-CAM - Seeing What the AI Sees**

This was probably the coolest part. Grad-CAM creates these heat map overlays that show exactly what parts of an image the AI is looking at when making decisions. What I found out:

* The AI focuses on areas with high contrast and sharp edges
* It cares about texture patterns (and apparently dragon scales look like cat fur to it?)
* It doesn't look at images the same way humans do at all

The heat maps were super useful for understanding why the AI makes the choices it does, even when those choices are completely wrong.

**Part 2: Breaking the AI (On Purpose)**

**The Experiment**

I wanted to see what would happen if I covered up the parts of the dragon image that the AI thought were most important for its (incorrect) cat classification. I tried three different ways:

1. **Black boxes** - just covered areas with solid black rectangles
2. **Heavy blur** - made important areas blurry but kept the colors
3. **Random noise** - added a bunch of random pixels to mess things up

**What Happened**

* **Original dragon (classified as "cat"):** 72.34% confidence
* **Black boxes:** 40.00% confidence (dropped by 32.34%)
* **Heavy blur:** 53.43% confidence (dropped by 18.91%)
* **Random noise:** 47.78% confidence (dropped by 24.56%)

Even when I was messing with a completely wrong classification, the occlusion still worked! The AI became less confident in its wrong answer when I blocked the areas it thought were important. It's like the AI was saying, "I'm not as sure this dragon is a cat anymore, but it's still some kind of cat."

**Part 3: Making Cool Filters**

**The "Deep Fried" Filter**

This was the most fun part! I created a filter that makes images look like those over-the-top memes you see online. Here's what it does:

1. Cranks up color saturation to 300%
2. Boosts contrast to 250%
3. Adds fake compression artifacts
4. Throws in some random noise for grittiness
5. Gives everything a red/orange tint
6. Sharpens everything aggressively for that "crispy" look

The result is these hilariously oversaturated, crunchy-looking images that look like they've been deep-fried in editing software. My dragon looked ridiculous after this filter - probably even more cat-like to the AI somehow.

**Other Filters I Made**

* **Blur:** Makes everything soft and dreamy
* **Edge Detection:** Shows just the outlines and boundaries
* **Sharpening:** Makes details pop out more
* **Emboss:** Creates this cool 3D carved effect

Each filter shows different aspects of the same image, which is pretty neat when you think about it.

**What I Learned**

**About Working with AI**

**The Good Stuff:**

* AI explanations were super helpful for understanding complex code
* It could break down big concepts into smaller, manageable pieces
* Good at generating working code with good comments

**The Reality Check:**

* You still need to understand what's going on to fix problems
* Sometimes there are compatibility issues (like Python version problems)
* High confidence doesn't mean the AI is right (my dragon is a very confident Egyptian cat!)

**The Dragon-Cat Discovery**

This was probably the biggest lesson of the whole project. AI classification systems don't have an "unknown" category - they HAVE to classify everything as one of their known objects, even when it makes no sense. My dragon became a cat because:

1. The AI was trained on a specific set of 1000 object categories
2. It doesn't have "dragon" or "mythical creature" as options
3. Maybe the scales and texture patterns reminded it of cat fur somehow?
4. It would rather be confidently wrong than admit it doesn't know

In real-world applications, this could be a serious problem. Imagine if this were a medical AI or something used for security - being confidently wrong could have real consequences.

**Technical Challenges**

I ran into some annoying Python version issues with TensorFlow – apparently, my Python 3.13 doesn't play nice with TensorFlow's Python 3.12 requirement. Had to get creative with workarounds using other libraries like PIL and NumPy.

**What This All Means**

This project showed me that AI is both more powerful and more limited than I initially thought. It can do amazing things with image recognition and processing, but it also has some pretty significant blind spots. The dragon-cat situation taught me that you can't just trust AI classifications blindly, especially when dealing with anything that might be outside the training data.

The combination of understanding how the AI works, testing its limits (and finding some pretty funny failures), and creating something creative with image processing gave me a much better appreciation for both the technical side and the artistic possibilities of computer vision.

Plus, I now know that AI thinks dragons are Egyptian cats, which is either concerning or hilarious depending on how you look at it.

**Check out all my code here:** <https://github.com/GyratGoldenwing/PythonProjects/tree/main/3.3%20Image%20Classification%20and%20Processing>