Image Classification & Processing Project

Final Report

Student: Kasey Carver
Date: January 22, 2026
GitHub Repository: https://github.com/KJCarver91/Artificial_Intelligence_Ecosystem

---

Part 1: Image Classification and Grad-CAM

Classification Results

Using a pre-trained MobileNetV2 model, I classified a stylized image of a tiger sitting at a classroom desk with the following results:

Top-3 Predictions:

1. Tiger Cat (64%)
2. Tiger (24%)
3. Home Theater (1%)

Analysis: The model's uncertainty between "tiger_cat" (64%) and "tiger" (24%) is particularly interesting given the image context. The image depicts a tiger in an unusual setting - sitting upright at a classroom desk in an anthropomorphized pose. This non-natural context likely confused the classifier, as the model was trained primarily on animals in typical environments.

The indoor classroom setting with furniture and books introduced visual elements associated with domestic spaces, which may explain why "tiger_cat" received higher confidence than "tiger." The crazy "home_theater" prediction (1%), while low confidence, demonstrates how environmental context can influence classification even when the primary subject is clearly identifiable. This highlights an important limitation: models trained on standard datasets may struggle with creative or unusual image compositions.

Grad-CAM Heatmap Analysis

The Grad-CAM visualization revealed how the neural network makes classification decisions:

Key Findings:

- **Strongest activation** appeared on the tiger's face, especially the eyes, nose, and facial structure
- **Moderate activation** on the body and stripe patterns
- **Minimal activation** on background elements

What I Learned: The heatmap shows the model focuses on biologically relevant features for identification, similar to how humans recognize animals. However, the unusual context of the image - a tiger sitting at a classroom desk - created additional challenges for the classifier. The concentration on facial features and stripe patterns demonstrates the network learned meaningful patterns, but the mixed visual cues from the indoor classroom environment (desks, chairs, chalkboard) likely contributed to the classification uncertainty between "tiger_cat" and "tiger."

This visualization reveals an important insight: while the model correctly identified feline features, the non-natural setting introduced contextual confusion. This proves the importance of understanding not just what the model sees, but how environmental context influences its predictions.

Technical Challenge: I resolved module import errors and ensured all image processing functions used TensorFlow's built-in operations (tf.nn.relu, tf.nn.softmax) for consistency with the MobileNetV2 implementation.

Part 2: Wild Neon Filter

Filter Design

I created a "Wild Neon" filter with these enhancements:

- Color saturation: 1.8x (vibrant colors)
- Contrast: 1.4x (dramatic depth)
- Sharpness: 2.0x (crisp details)
- Edge enhancement: 30% blend (subtle glow)

Effect on Images

The filter transforms images by:

- Making colors more vivid while maintaining natural relationships
- Enhancing texture details like fur and whiskers

- Creating dramatic contrast that makes subjects pop from backgrounds
- Adding a contemporary glow effect around edges

This filter is ideal for social media posts, wildlife portfolios, and creative projects requiring bold visual impact.

Part 3: AI-Assisted Development Reflection

What Worked Well

Code Explanation: The AI provided clear explanations that went beyond definitions to explain *why* certain approaches were used, deepening my understanding.

Debugging: When I encountered errors, the AI quickly identified root causes, provided specific fixes, and explained the reasoning. This turned frustration into learning opportunities.

Creative Collaboration: Describing the visual effect I wanted led to specific technical implementations with adjustable parameters, demonstrating AI's value for translating creative vision into code.

Challenges

- **Over-reliance risk:** I had to consciously verify I understood code rather than just copying it
- **Context assumptions:** The AI sometimes assumed incorrect environment details requiring clarification
- **Verification needed:** Running code incrementally was essential to catch issues early

Skills Developed

- Python image processing (PIL, OpenCV, TensorFlow, matplotlib)
- Systematic debugging methodology
- Git workflow (forking, branching, committing, pushing)
- Critical evaluation of AI suggestions
- Image processing concepts (filters, enhancement, edge detection)

Key Takeaway

AI assistance accelerates learning and problem-solving, but fundamental programming knowledge remains essential for evaluating suggestions critically. The AI is most valuable for learning new libraries, debugging complex errors, and exploring different approaches - but understanding core concepts allows me to adapt solutions appropriately.

Conclusion

This project demonstrated practical applications of deep learning for image classification, the importance of model interpretability through Grad-CAM, and creative possibilities in image processing. AI assistance transformed the learning experience into an interactive collaboration, helping me understand both the "how" and "why" of computer vision techniques.