Brandy Griffin

Department of Science, Technology, Engineering & Math, Houston

ITAI 1378 Computer Vision

Fall 2024

Professor Patricia McManus

Wednesday, October 09, 2024

**Chihuahua vs Muffin CNN**

**Introduction: Choosing Google Colab**

For this assignment, students were given two options: AWS SageMaker Studio Lab or Google Colab. I opted to use Google Colab due to its simplicity in setup and the convenience of running everything in the cloud without requiring any installations. The command I used to clone the project was:

!git clone <https://github.com/patitimoner/workshop-chihuahua-vs-muffin.git>

Once that was done, I was able to move forward with the task.

**CNN Architecture**

I learned how Convolutional Neural Networks (CNNs) work and why they’re great for image classification. Unlike traditional neural networks, CNNs focus on small sections of an image to recognize patterns like edges and shapes, which is useful for telling apart similar things, like chihuahuas and muffins. The model we built had layers that first identified simple features like edges, then more complex shapes. However, unlike traditional networks, I couldn’t see the images during classification and had to rely on performance metrics like accuracy, which made it harder to track progress visually.

**Model Performance**

I encountered several challenges during the training process. For instance, I needed to ensure the data was loaded correctly, and I had to resolve some issues in steps 3 to 5 and step 7.

When I received the error messages, I used the built-in Gemini tool in Google Colab to help me understand what went wrong. Gemini gave me useful suggestions for fixing my code, which made it easier to troubleshoot. One of the biggest challenges was that the CNN didn't show the images during training, so I had to rely on accuracy scores and loss values to see how well the model was doing.

**CNN vs. Traditional Neural Network**

A convolutional neural network (CNN) is a type of neural network that uses special layers called convolutional layers. While regular neural networks are often compared to the human brain (though that’s not entirely accurate), CNNs are more inspired by the visual system in humans and animals, which is a bit closer to the truth. The convolutional layer applies something called a convolution, using a filter that it learns as it processes the data. This helps the network detect patterns in images, like edges, corners, and eventually more complex shapes. CNNs usually also include other layers, like pooling and dense layers.

When comparing CNNs to traditional neural networks, one challenge I faced was the lack of visual feedback during the classification process. In traditional neural networks, I could see the images as they were being classified, making it easier to evaluate the model's performance. With CNNs, however, I had to depend more on performance metrics and focus on how the model was learning internally, which made it harder to interpret the results at first. Without the ability to see the images during each step, it was more difficult to understand how well the model was working in the beginning stages.

**Challenges and Solutions**

One of the main challenges I faced was handling the errors that came up while setting up the data and running the model. For example, in steps 3 to 5 and again in step 7, I had to check that the file paths were correct, and that the data was properly transformed into a format that the model could use. The Gemini tool in Google Colab was really helpful for resolving these issues. The tool helped explain the error messages and made suggestions on ways to fix them, which made debugging easier.

Another challenge was the lack of visual output from the CNN model. Since I couldn't see the images while the model was running, I had to rely on the accuracy and loss values to see how well it was doing. This was different from traditional neural networks, where I could visually inspect how the model was performing by looking at the images and the results.

**Real-World Applications**

Convolutional Neural Networks (CNNs), especially in face recognition, have lots of real-world uses across different industries. On social media platforms like Facebook, face recognition powered by CNNs makes it easier to tag friends in photos, saving time when going through large albums. Outside of social media, CNN-based face recognition is also being used in security for identifying people, like at airports or secure buildings, to verify their identity. It’s even used in surveillance systems to help law enforcement track people in real-time through security cameras. Plus, face recognition with CNNs is becoming common in consumer devices like smartphones and laptops, where it’s used for things like unlocking the device or accessing secure apps with just your face.

**Ethical Considerations**

The use of Convolutional Neural Networks (CNNs) in face recognition raises important ethical concerns. Privacy is a major issue, as facial images can be collected without consent, especially in public surveillance, leading to potential privacy violations and mass surveillance. Bias is another concern; if CNNs are trained on non-diverse data, they may misidentify or discriminate against certain groups, like people of color or those with unique features. Additionally, face recognition isn’t always as accurate as methods like fingerprints, and mistakes in critical areas like law enforcement could have serious consequences. It’s essential to address these concerns to ensure fair and responsible use of the technology.

**Conclusion**

This assignment taught me a lot about how CNNs work and how they can be applied to real-world problems. Choosing Google Colab made it easier to run the code, and even though I ran into challenges with file paths and data transformations, the Gemini tool helped me troubleshoot and fix the issues.

I also learned that CNNs focus more on the training process and don't show visual outputs the way traditional neural networks do. This forced me to pay more attention to the performance metrics and understand the model in a different way.

**References**

Mukhammadiev, Aytan. "Convolutional Neural Network Explained in 7 Real-Life Examples." *Medium*, 22 Aug. 2020, <https://medium.com/swlh/convolutional-neural-network-expalained-in-7-real-life-examples-6015a64f9d2a>.

<https://colab.research.google.com/github/brandiegriffin83/brandiegriffin83/blob/main/CNN_1%20Chihuahua%20or%20Muffin.ipynb>