**Christian Mpabuka**

**Lab 07 - Chihuahua or Muffin Workshop Instructions**

**ITAI 1378 Computer Vision**

**Professor: Patricia McManus**

**Introduction**

In my journal, I will reflect on my experience with the CNN-based classification task. This task marked a significant advancement from the traditional neural network tasks we had previously worked on. I will delve into the CNN architecture, its distinctions from traditional neural networks, and share my personal experiences and insights gained during this task.

I've chosen Option A for this lab because I'm familiar with Google Colab. I'll walk through each step of the lab, discussing the challenges I faced and the steps I took to overcome them.

First step of the Lab I cloned the Lab by using the code given in the assignment.

!git clone https://github.com/patitimoner/workshop-chihuahua-vs-muffin.git  
 %cd workshop-chihuahua-vs-muffin  
  !ls

Step 2: As I ran the code, I encountered errors. I attempted to troubleshoot and fix the errors on my own before seeking assistance from Gemini. After making the necessary corrections to the code, the output matched the expected results, and any incorrect predictions were rectified.

**CNN Architecture:**

Convolutional Neural Networks (CNNs) are a type of deep neural network that are highly effective for analyzing visual data. Unlike traditional neural networks, CNNs are specifically designed to automatically and adaptively learn spatial hierarchies of features through backpropagation. They achieve this by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers

**Convolution Layers:** These layers apply a convolution operation to the input, passing the result to the next layer. This operation helps in detecting various features in the image, such as edges, textures, and patterns.

**Pooling Layers:** These layers perform a down-sampling operation along the spatial dimensions (width, height), reducing the dimensionality of the feature map and helping to make the detection of features invariant to scale and orientation.

**Fully Connected Layers:** These layers are like those in traditional neural networks and are used to make predictions based on the features extracted by the convolution and pooling layers.

**Personal Experience**

Working with CNNs was both challenging and rewarding. Initially, understanding the convolution operation and how it differs from the operations in traditional neural networks was a bit daunting. However, through hands-on practice and visualizing the feature maps, I gained a deeper appreciation for the power of CNNs in image classification tasks.

One of the most enlightening moments was when I saw the model’s performance improve significantly after tuning the hyperparameters and adding more convolutional layers. This experience underscored the importance of model architecture and the impact of each layer on the overall performance.

**Challenges and Solutions**

**Challenge:** Understanding and correctly implementing the convolution and pooling operations

**Solution:** I addressed this with tools such as Gemini and ChatGPT, including tutorials and documentation, and practicing with small examples to grasp the concepts better

**Model Performance**

The model’s performance was evaluated based on accuracy and the patterns observed in misclassifications. CNN achieved an accuracy of 100% on the test dataset.

**Accuracy:** The model showed high accuracy in classifying images with clear and distinct features.

**Comparison**

In comparing the CNN with the traditional neural network model, we found significant differences in performance and training time.

**Performance:** The CNN outperformed the traditional neural network in terms of accuracy. The traditional neural network achieved an accuracy of [insert 100% compared to the CNN’s [insert accuracy]%. This improvement is attributed to CNN’s ability to capture spatial hierarchies and local features more effectively.

**Training Time:** CNN required more training time due to its complex architecture and the need for more computational resources. However, the improved performance justified the additional training time.

**Real-World Applications**

CNNs, or Convolutional Neural Networks, have a wide range of real-world applications, particularly in the field of image and video analysis. Some potential applications include:

1. Medical Imaging: CNNs can be used to detect and diagnose diseases from medical images such as X-rays, MRIs, and CT scans.

2. Autonomous Vehicles: CNNs are crucial for object detection and recognition in self-driving cars, helping them navigate and make decisions based on their surroundings.

3. Facial Recognition: CNNs are widely used in security systems for identifying individuals based on facial features.

4. Retail: CNNs can be used for visual search and recommendation systems, enhancing the shopping experience by allowing customers to search for products using images.

**Conclusion**

The CNN-based classification task was a valuable learning experience that deepened my understanding of deep learning and its applications in image processing. It also provided practical insights into the challenges and solutions associated with training complex models. I am looking forward to applying these learnings to future projects and continuing to explore the capabilities of CNNs..