# Chihuahua vs Muffin Experiment Journal

**Introduction:**

In this project, I trained a Convolutional Neural Network (CNN) to tell the difference between pictures of Chihuahuas and muffins. The idea was to try different experiments and see how changing the model, the learning rate, or the number of layers affects the accuracy. I did six experiments in total, each testing one specific thing.

**Experiment 1:**

In the first experiment, I used the simple version of the model with the default settings: learning rate 0.001 and 10 training epochs. This experiment helped me understand how the model works without any changes. The accuracy was good enough to start with. It showed me what the model could do on its own without tuning. This experiment was used as a reference to compare with the others.

**Experiment 2:**

In this experiment, I reduced the learning rate from 0.001 to 0.0001. I wanted to see what happens when the model learns more slowly. The model improved very slowly, and the accuracy did not change much. It learned, but it was too slow. This helped me see that a learning rate that is too small makes training take longer and sometimes does not give better results.

**Experiment 3:**

For this experiment, I kept everything the same as the baseline but trained the model for 20 epochs instead of 10. The model did better because it had more time to learn. But after a while, the improvement started slowing down. This showed me that training longer can help, but only up to a point.

**Experiment 4:**

Here, I removed one convolutional layer from the model (so instead of 3 layers, it had only 2). The model became simpler, and as expected, it did not perform as well. The accuracy dropped. This experiment showed me that having more layers helps the model learn more details from the images.

**Experiment 5:**

In this experiment, I added an extra convolutional layer (4 instead of 3). The model became deeper and learned more complex features. This increased the accuracy compared to the

baseline. The model understood the images better, but training also took a little longer. This experiment showed that deeper models can perform better when trained correctly.

**Experiment 6:**

Here, I changed the learning rate from 0.001 (baseline) to 0.01. The idea was to see what happens if the model learns too fast. The result was not good. The accuracy went down because the model started making bigger mistakes during training. This experiment showed me that a learning rate that is too high can make the model unstable.

**Challenges and Solutions:**

During this project, I faced a few challenges. One challenge was understanding why some experiments gave better results while others made the accuracy drop. At first, this was confusing because small changes in the learning rate or layers sometimes caused big differences. I solved this by comparing each experiment carefully and checking how the model behaved during training. Another challenge was dealing with errors in the code, especially when some functions were missing or not defined yet. To fix this, I checked my notebook step by step and made sure everything was run in the correct order. Over time, I got better at reading the results and understanding the model.

**Real World Applications:**

This kind of image classification model can be used in many real situations. For example, it could help in animal shelters to identify dog breeds. It could also be used in food apps to recognize food items from photos. The same model structure can even be used in medical imaging or security cameras. This project shows how machines can learn to see patterns that people sometimes confuse.

**Conclusion:**

From these experiments, I saw how different choices affect a CNN model. A good learning rate and enough layers help the model perform well. Training longer usually helps, but only up to a point. Too many layers or too high a learning rate can cause problems. Overall, I learned how CNNs work and how small changes can make a big difference.