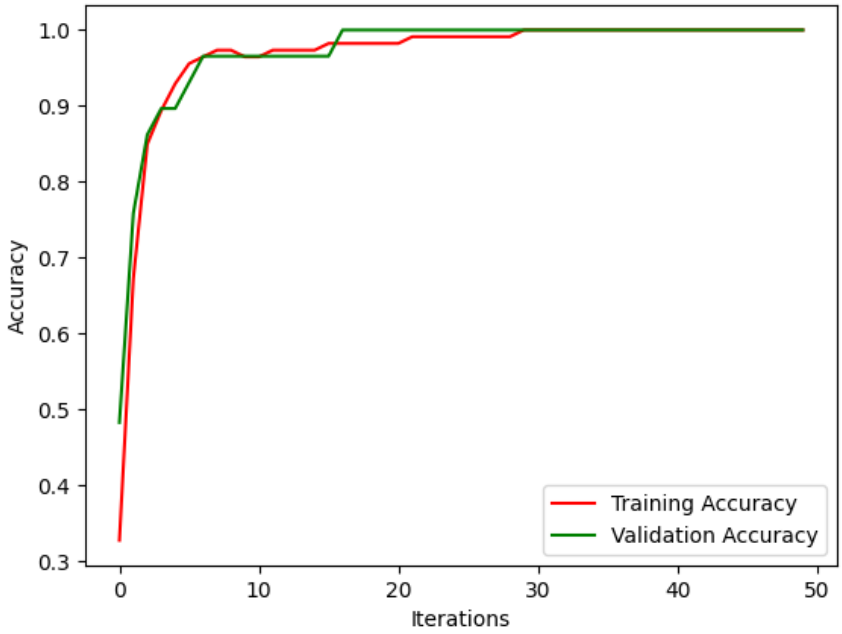
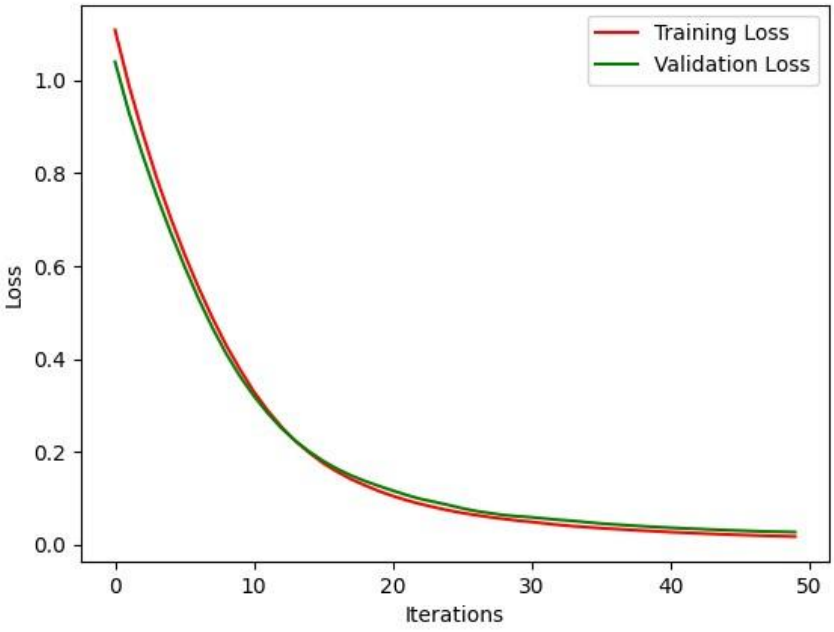
Problem 1:

Accuracy vs. Iterations



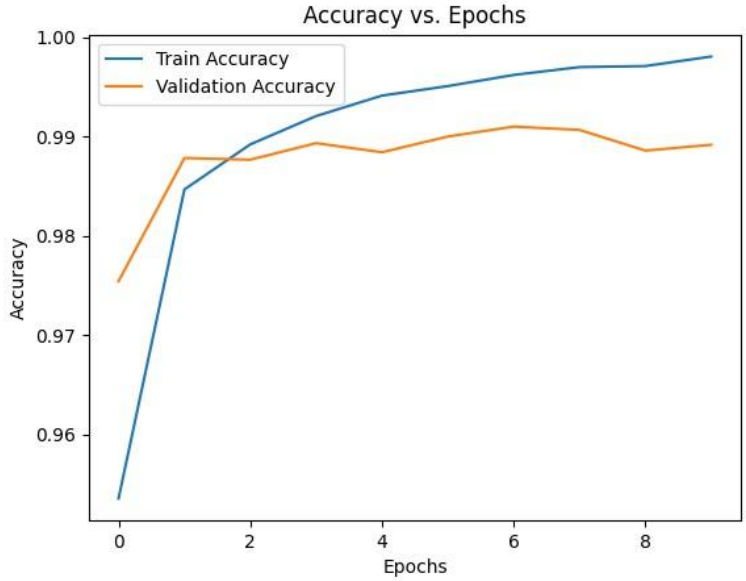
Loss vs. Iterations

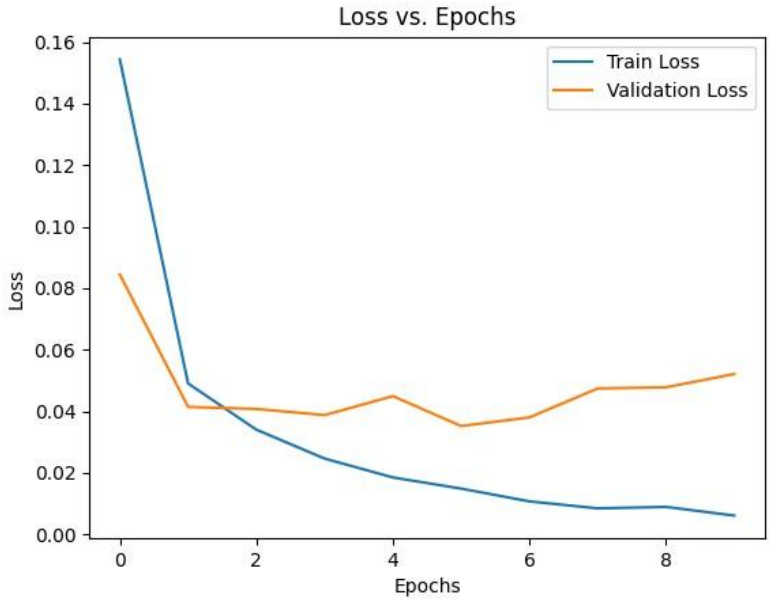


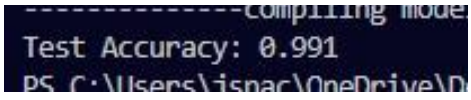
Accuracy Score

```
training model...
Test Accuracy: 1.000
PS C:\Users\jspac\OneDrive\Desktop>
```

Problem 2:



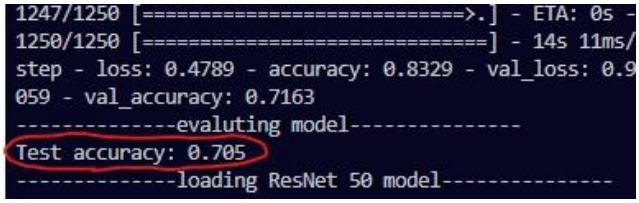


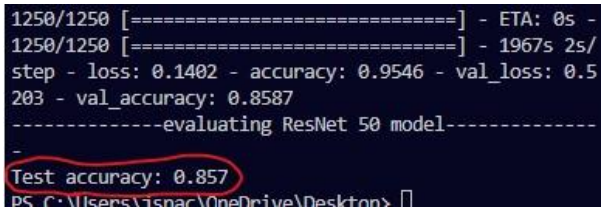


My choice of layers:

For this CNN, I implemented two convolutional layers, the first with 32 filters and the second with 64 filters. Then, I used two max pooling layers to reduce the spatial dimensions of the feature maps in order to prevent overfitting and reduce computational expense, while keeping the more important features intact. Then, I used a flatten layer to reshape the two-dimensional feature maps into a one-dimensional vector so that it can be fed into the fully connected layers.

Problem 3:





Accuracy of my model: **0.705**
Accuracy of ResNet50: **0.857**

% difference $= \frac{0.857-0.705}{\frac{0.857+0.705}{2}} \times 100 = 19.46\%$

ResNet50 is 19.46% more accurate

References:

https://www.tensorflow.org/tutorials/images/transfer_learning
https://keras.io/api/applications/
https://keras.io/api/applications/#usage-examples-for-image-classification-models
https://towardsdatascience.com/deep-learning-using-transfer-learning-python-code-for-resnet50-8acdfb3a2d38
https://www.w3schools.com/python/matplotlib_pyplot.asp
https://www.geeksforgeeks.org/python-introduction-matplotlib/
https://keras.io/examples/vision/mnist_convnet/
https://medium.com/@kenneth.ca95/a-guide-to-transfer-learning-with-keras-using-resnet50-a81a4a28084b