Problem 1

The focus of this paper is to discuss how easily deep neural networks can be fooled into misidentifying images. It explains that it is very easy to create an image that a DNN identifies very confidently as a thing but makes absolutely no sense to human eyes. They generated images with evolutionary algorithms and then trained DNNs on them. With the MNIST data set, there were similar results (99.99% confidence) whether the images were irregular static or more regular images. A similar test was done with the ImageNet set, and while they thought more data would mean less error, the models still produced errors.

I liked looking through all the pictures they included. For some, I could see how they could be identified the way they were, but for others they made no sense whatsoever. It was fun to go through them and try and figure out what they were supposed to be or guess why they were identified how they were. I also liked the discussion of the paper and found it interesting that their original goal was far from what had happened.

There wasn't much I disliked. The only thing I can think of was I thought the occasional "see this website for more" or "go look for this in supplementary materials" threw off the flow of reading it, but that's just me being picky.

When I saw the topic of the paper, it had my interest because it reminded me of software that I've used before but didn't fully understand. It's called Glazed, and it's described as a "cloaking tool" to prevent AI models from replicating the style of the cloaked image. It's used by artists to protect against people running their work through AI to copy them. To a person, it just adds a sort of grainy or wavy looking layer to it. It was the first thing I thought of when I saw some of the images in the article. There's another software that works even more closely with what's described in the article called Nightshade. That one adds just enough distortion so a person can barely notice, but an AI mislabels the image entirely. After reading the article, I might end up playing around with each software and seeing what it can do.

Problem 2

I had no installation issues.

Problem 3

The original settings had it below the .95 mark, so I started tweaking the number of epochs first. 15 helped, and then I bumped up batch size to 72 to see what happened. Training accuracy was near perfect (.998) and validation accuracy held around .982, averaging ~.95 overall in the summary. When I tried 20 epochs, performance improved slightly, with precision reaching .96. However, changing other parameters didn't help much. I tried a 0.25 train- split, which dragged results down to ~.93, and lowering the batch size to 32 kept things stuck at ~.95. It seemed like it would be hard to push higher numbers.

My most successful run was when I increased to 25 epochs and a batch size of 72. Training loss fell to 0.0015, accuracy hit 0.9995, and validation accuracy rose a bit to 0.9843 with lower validation loss (0.0966). This was the best balance I found. Running 50 epochs (while I got dinner) pushed train accuracy to 1.0 and nearly zero loss, but validation accuracy barely moved (0.9874) while runtime nearly quadrupled, making it not worth it in my mind.