Generation of a large-scale model of the retina using AI

Kumar Mrinal

January 29, 2025

1 Introduction

How do neural circuits transform the chaos of the natural world into meaningful signals for the brain? This is one of neuroscience's greatest mysteries, and the retina—a gateway to visual processing—holds the answers. Packed with over 50 types of interneurons, the retina is like a biological supercomputer, evolved to decode natural scenes. But here's the catch: nearly everything we know about it comes from artificial experiments, like flashing dots and drifting patterns, which barely scratch the surface of real-world complexity.

Enter AI and Machine Learning (AIML). By harnessing the power of deep learning, three-layer convolutional neural networks (CNNs) have revolutionized how we study the retina. These models don't just mimic retinal responses to natural scenes—they reveal the intricate pathways behind motion detection, adaptation, and prediction, bridging the gap between biology and computation. Even cooler? They unlock new hypotheses about how interneurons perform their magic, offering a fresh lens to decode the natural world. AIML isn't just a tool—it's the key to cracking the retina's code.

2 Conclusion

By designing a minimal three-layer CNN inspired by the retina's structure, paper captured the complex neural code for natural scenes with remarkable precision. Mimicking the retina's three levels of rectification, the model replicated the responses of salamander retinal ganglion cells to natural images and white noise. With tiled convolutional layers and spatially limited inputs, this CNN revealed how intricate retinal computations arise from structured, interpretable pathways. Optimized with gradient descent, the model not only decoded retinal outputs but also opened the door to understanding the hidden mechanisms of natural visual processing—a leap forward in neuroscience and AI.

Dockerfile for Torch-Deep-Retina

Below is the `Dockerfile` used to set up the environment for the `torch-deep-retina` project:

```
# Use Python 3.9 slim as the base image
FROM python:3.9-slim

# Set the working directory inside the container
WORKDIR /app

# Copy the entire project into the container
COPY . /app

# Install Python dependencies
COPY baccuslab/torch-deep-retina-v1.0/baccuslab-torch-deep-retina-9
    f367b1/requirements.txt /app/requirements.txt
RUN pip install --upgrade pip && pip install -r /app/requirements.
    txt

# Set the default working directory for further commands
WORKDIR /app/baccuslab/torch-deep-retina-v1.0/baccuslab-torch-deep-
    retina-9f367b1/training_scripts

# Set the entry point for running scripts
CMD ["python3"]
```

Listing 1: Dockerfile Setup

Execution Commands for Torch-Deep-Retina

Below are the commands used to build, run, and execute the training script for the `torch-deep-retina` project:

Building the Docker Image

```
docker build -t torch-deep-retina .
```

Listing 2: Docker Build Command

Running the Docker Container

```
docker run -it --rm --gpus all torch-deep-retina /bin/bash
```

Listing 3: Docker Run Command

Navigating to Training Scripts Directory

```
cd /app/baccuslab/torch-deep-retina-v1.0/baccuslab-torch-deep-
    retina-9f367b1/training_scripts
```

Listing 4: Directory Navigation Command

Executing the Training Script

```
CUDA_VISIBLE_DEVICES=0 python3 main.py "C:\Users\krmri\Downloads
    \8001612\baccuslab\torch-deep-retina-v1.0\baccuslab-torch-deep-
    retina-9f367b1\training_scripts\hyperparams.json"
"C:\Users\krmri\Downloads\8001612\baccuslab\torch-deep-retina-v1.0\
    baccuslab-torch-deep-retina-9f367b1\training_scripts\
    hyperranges.json"
```

Listing 5: Training Script Execution

Illustrations of the Torch-Deep-Retina Project

Below are some key visuals demonstrating the attempt at the project:

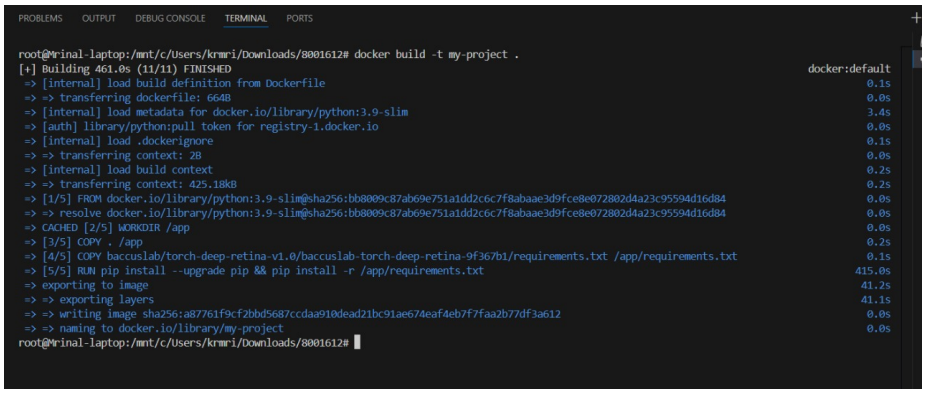

Figure 1:

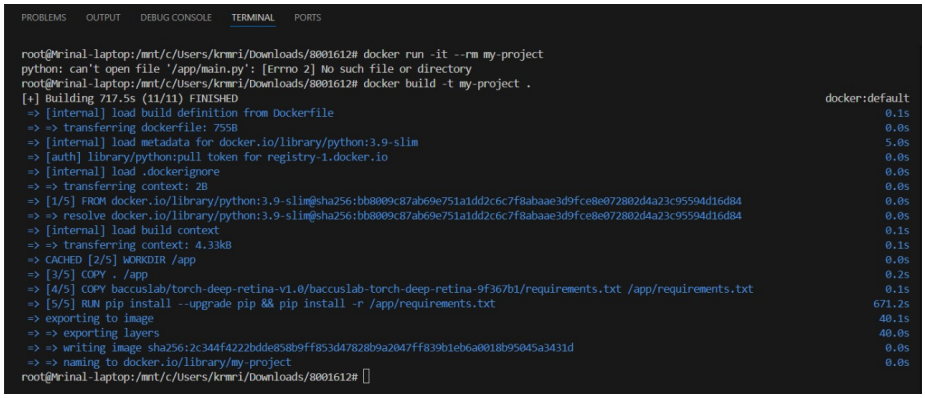

Figure 2:

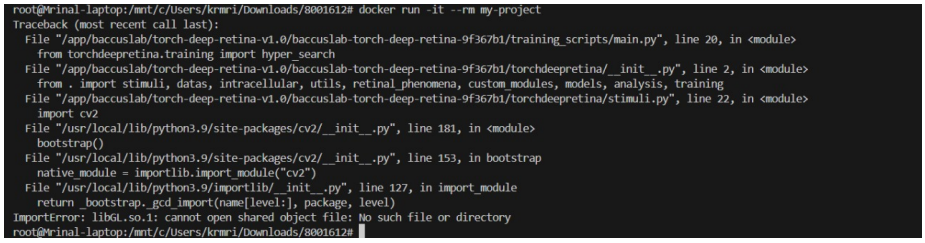

Figure 3:

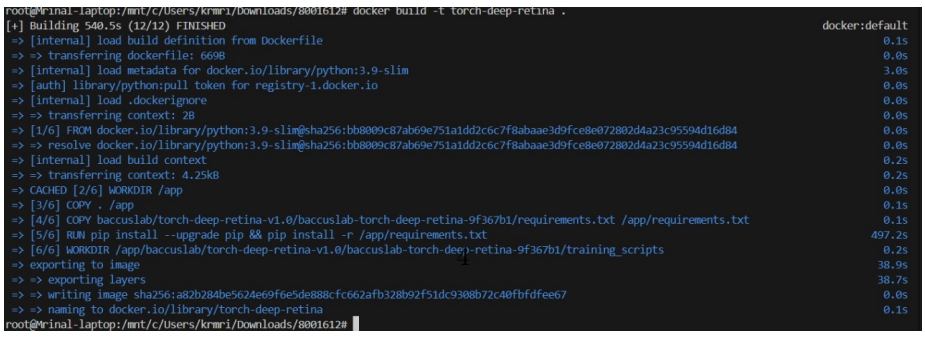

Figure 4: