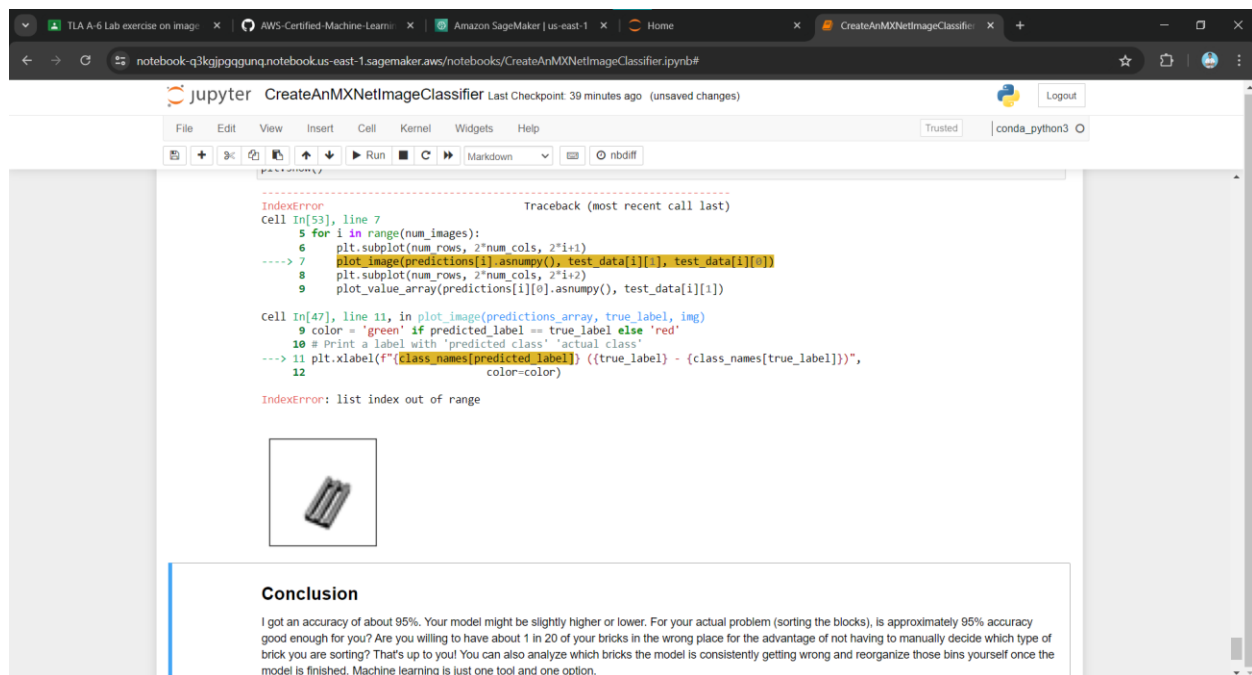
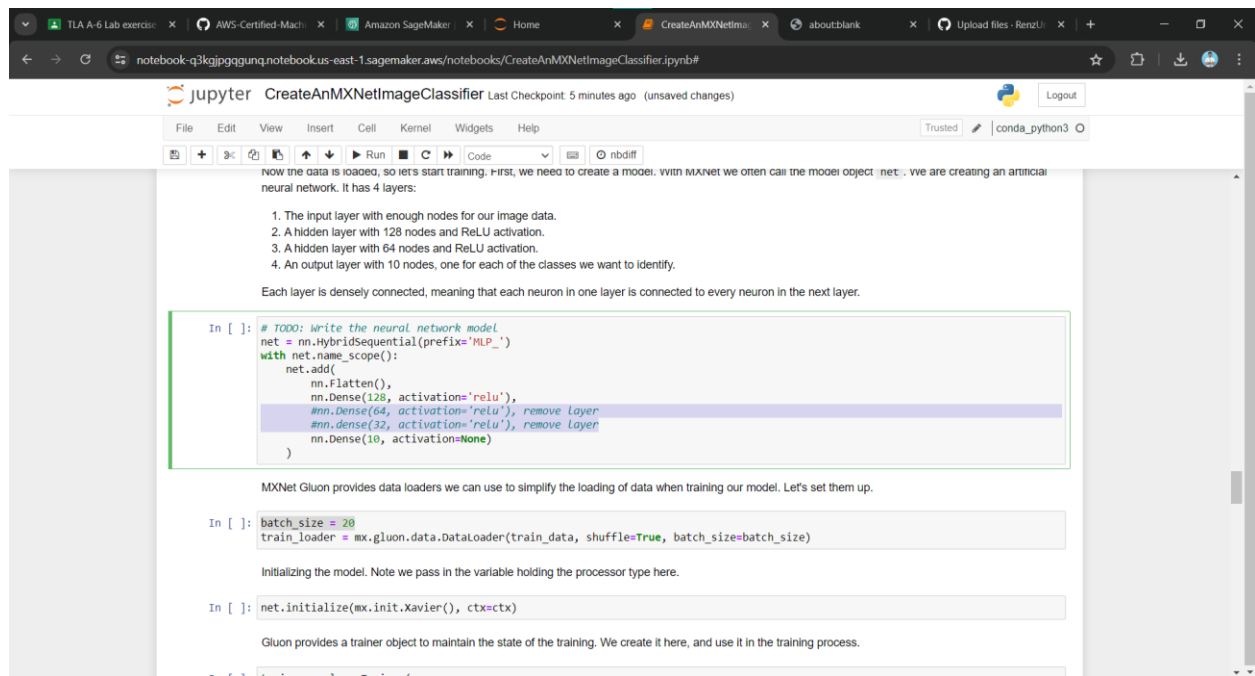


Added another hidden layer



By adding an additional layer, there errors that can be found on the code and there is no graph being presented to visualize the accuracy of the data. Nonetheless, the output is different which in this case is the LEGO Grill.



notebook-q3kjpgggunq.notebook-us-east-1.sagemaker.aws/notebooks/CreateAnMXNetImageClassifier.ipynb#

jupyter CreateAnMXNetImageClassifier Last Checkpoint: 5 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted conda\_python3

now the data is loaded, so let's start training. First, we need to create a model. Within MXNet we often call the model object `net`. We are creating an artificial neural network. It has 4 layers:

1. The input layer with enough nodes for our image data.
2. A hidden layer with 128 nodes and ReLU activation.
3. A hidden layer with 64 nodes and ReLU activation.
4. An output layer with 10 nodes, one for each of the classes we want to identify.

Each layer is densely connected, meaning that each neuron in one layer is connected to every neuron in the next layer.

```
In [ ]: # TODO: Write the neural network model
net = nn.HybridSequential(prefix='MLP_')
with net.name_scope():
    net.add(
        nn.Flatten(),
        nn.Dense(128, activation='relu'),
        #nn.Dense(64, activation='relu'), remove layer
        #nn.dense(32, activation='relu'), remove layer
        nn.Dense(10, activation=None)
    )
```

MXNet Gluon provides data loaders we can use to simplify the loading of data when training our model. Let's set them up.

```
In [ ]: batch_size = 20
train_loader = mx.gluon.data.DataLoader(train_data, shuffle=True, batch_size=batch_size)
```

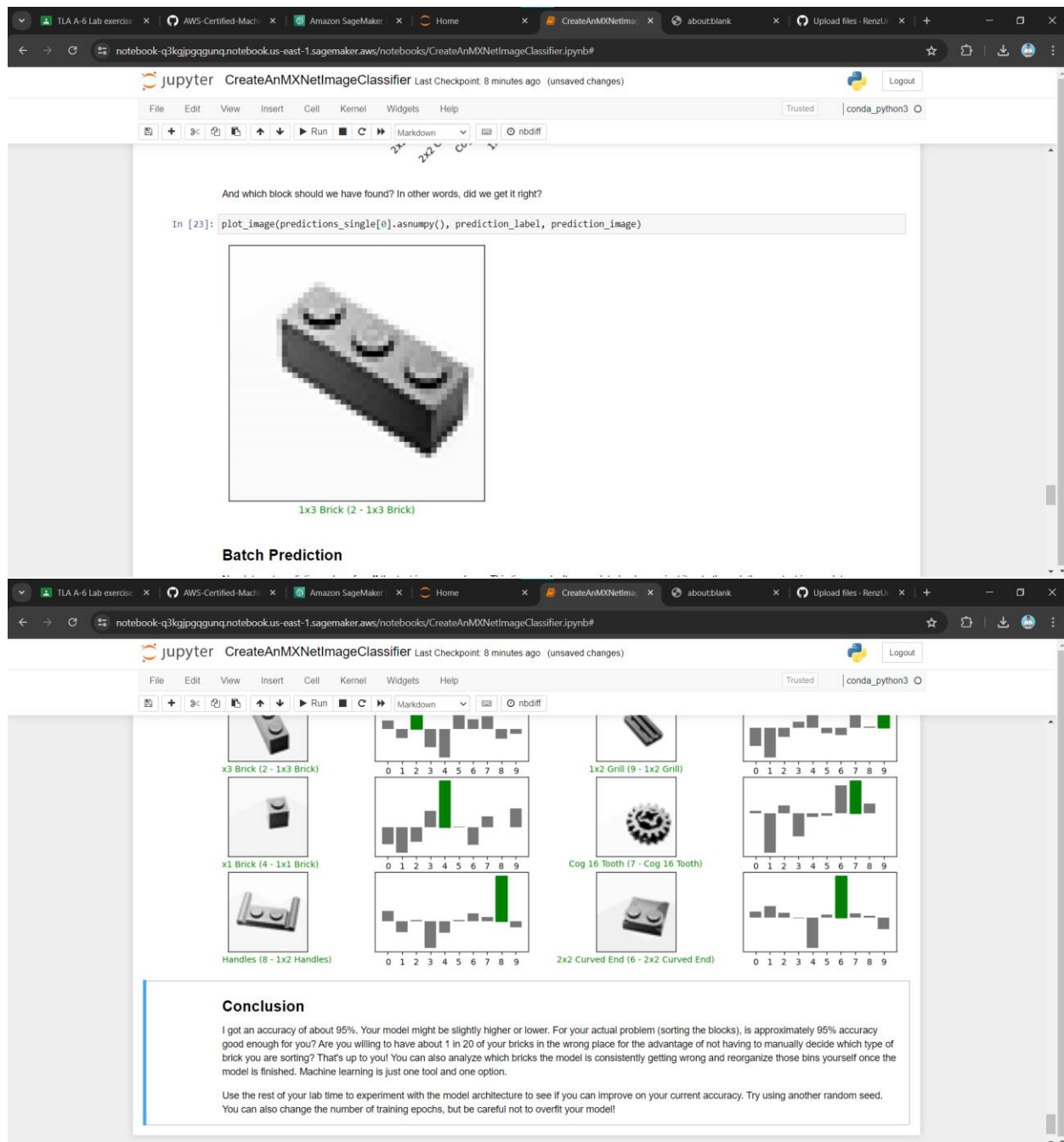
Initializing the model. Note we pass in the variable holding the processor type here.

```
In [ ]: net.initialize(mx.init.Xavier(), ctx=ctx)
```

Gluon provides a trainer object to maintain the state of the training. We create it here, and use it in the training process.

```
In [ ]: trainer = gluon.Trainer()
```

Removed two layers



The output is quite similar if the two layers are not removed especially when it comes to the accuracy and the type of Lego brick output.