```
pip install datasets
```

```
Collecting datasets
  Downloading datasets-3.6.0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: filelock in
/usr/local/lib/python3.11/dist-packages (from datasets) (3.18.0)
Requirement already satisfied: numpy>=1.17 in
/usr/local/lib/python3.11/dist-packages (from datasets) (2.0.2)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.11/dist-packages (from datasets) (18.1.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in
/usr/local/lib/python3.11/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in
/usr/local/lib/python3.11/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in
/usr/local/lib/python3.11/dist-packages (from datasets) (4.67.1)
Collecting xxhash (from datasets)
  Downloading xxhash-3.5.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py311-none-any.whl.metadata (7.2
kB)
Collecting fsspec<=2025.3.0,>=2023.1.0 (from
fsspec[http]<=2025.3.0,>=2023.1.0->datasets)
  Downloading fsspec-2025.3.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: huggingface-hub>=0.24.0 in
/usr/local/lib/python3.11/dist-packages (from datasets) (0.30.2)
Requirement already satisfied: packaging in
/usr/local/lib/python3.11/dist-packages (from datasets) (24.2)
Requirement already satisfied: pyyaml>=5.1 in
/usr/local/lib/python3.11/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/usr/local/lib/python3.11/dist-packages (from
fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (3.11.15)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.24.0-
>datasets) (4.13.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.32.2-
>datasets) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.32.2-
>datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.32.2-
>datasets) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.32.2-
```

```
>datasets) (2025.4.26)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas->datasets)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.11/dist-packages (from pandas->datasets)
(2025.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas->datasets)
(2025.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.6.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (6.4.3)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]<=2025.3.0,>=2023.1.0->datasets) (1.20.0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2-
>pandas->datasets) (1.17.0)
Downloading datasets-3.6.0-py3-none-any.whl (491 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 491.5/491.5 kB 7.3 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 6.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 193.6/193.6 kB 6.4 MB/s eta
0:00:00
ultiprocess-0.70.16-py311-none-any.whl (143 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 2.8 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 9.9 MB/s eta
0:00:00
ultiprocess, datasets
  Attempting uninstall: fsspec
```

```
    Found existing installation: fsspec 2025.3.2
    Uninstalling fsspec-2025.3.2:
      Successfully uninstalled fsspec-2025.3.2
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2025.3.0
which is incompatible.
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cublas-cu12 12.5.3.2 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-cupti-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-nvrtc-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cuda-runtime-cu12 12.5.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cudnn-cu12 9.3.0.75 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cufft-cu12 11.2.3.61 which is incompatible.
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-curand-cu12 10.3.6.82 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cusolver-cu12 11.6.3.83 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-cusparse-cu12 12.5.1.3 which is incompatible.
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127;
platform_system == "Linux" and platform_machine == "x86_64", but you
have nvidia-nvjitlink-cu12 12.5.82 which is incompatible.
Successfully installed datasets-3.6.0 dill-0.3.8 fsspec-2025.3.0
multiprocess-0.70.16 xxhash-3.5.0

#---------------------------------IMPORT
LIBRARIES-------------------------------------

from datasets import load_dataset
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

```
import numpy as np

from sklearn.metrics import classification_report

import random
from collections import defaultdict

#------------LOAD DATASET AND SPLIT
TRAIN/TEST----------------------------------------

ds = load_dataset("uoft-cs/cifar100")# Load the CIFAR10 dataset
(train_images, train_labels), (test_images, test_labels) =
datasets.cifar10.load_data()
train_images, test_images = train_images / 255.0, test_images / 255.0
#Normalize pixel values to be between 0 and 1

/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/
_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your
settings tab (https://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to
access public models or datasets.
  warnings.warn(

{"model_id":"87b914cd13b842bebca31082b779819d","version_major":2,"vers
ion_minor":0}

{"model_id":"01908499cd0a482da1756a533d881d95","version_major":2,"vers
ion_minor":0}

{"model_id":"ffff5dfe2d104cb5a343f935fff891d5","version_major":2,"vers
ion_minor":0}

{"model_id":"fb3e65ba43db40da8dc7702ab1e06aab","version_major":2,"vers
ion_minor":0}

{"model_id":"f596eb7363f04244ab7571ef3fe84a17","version_major":2,"vers
ion_minor":0}

Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-
python.tar.gz
170498071/170498071 ━━━━━━━━━━━━━━━━━━━━ 4s 0us/step

#-------------DEFINE THE CNN MODEL
ARCHITECTURE-----------------------------------------------------

model = models.Sequential() #A Simple basic sequential model
model.add(layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(32, 32, 3)))  # Convolutional Layer by filtering 3x3
```

```
matrix - 32 kernel size; Input shape for CIFAR10= 32x 32 with RGB (3)-
Color
model.add(layers.MaxPooling2D((2, 2))) #Max pooling 2x2
model.add(layers.Conv2D(64, (3, 3), activation='relu')) #Activation
Function - relu for non-linearlity
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())  # Flatten the 3D feature maps
model.add(layers.Dense(64, activation='relu')) #64 kernels
model.add(layers.Dense(10, activation='softmax'))  # 10 classes for
CIFAR10
```

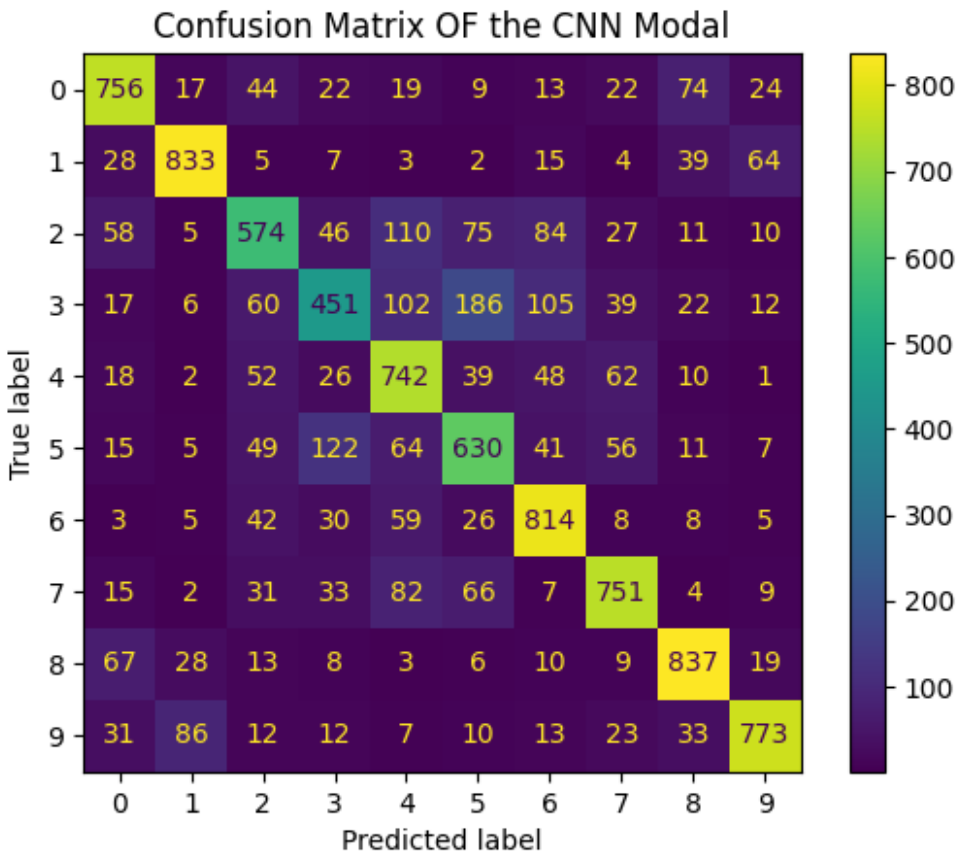
```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/
convolutional/base_conv.py:107: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)
```

```
#-----------------COMPILE
MODEL----------------------------------------------------------------
```

```
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',
metrics=['accuracy'])  #define loss function ,  metrices and
optimization
model.fit(train_images, train_labels, epochs=10,
validation_data=(test_images, test_labels))
test_loss, test_acc = model.evaluate(test_images, test_labels,
verbose=2)
print(test_acc)
```

```
Epoch 1/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 73s 45ms/step - accuracy: 0.3447 -
loss: 1.7596 - val_accuracy: 0.5348 - val_loss: 1.2841
Epoch 2/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 46ms/step - accuracy: 0.5874 -
loss: 1.1708 - val_accuracy: 0.6266 - val_loss: 1.0591
Epoch 3/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 46ms/step - accuracy: 0.6388 -
loss: 1.0299 - val_accuracy: 0.6474 - val_loss: 1.0267
Epoch 4/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 80s 44ms/step - accuracy: 0.6769 -
loss: 0.9200 - val_accuracy: 0.6651 - val_loss: 0.9406
Epoch 5/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 83s 45ms/step - accuracy: 0.7050 -
loss: 0.8379 - val_accuracy: 0.6722 - val_loss: 0.9491
Epoch 6/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 72s 46ms/step - accuracy: 0.7282 -
loss: 0.7747 - val_accuracy: 0.6931 - val_loss: 0.8744
```

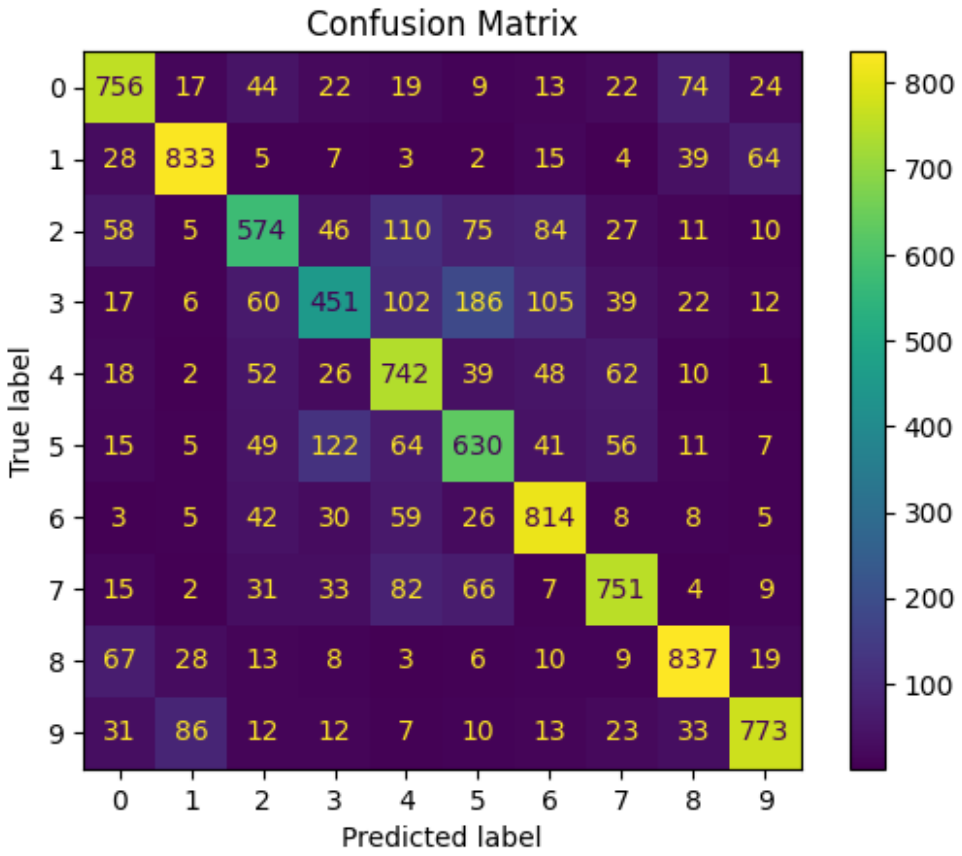
```
Epoch 7/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 73s 46ms/step - accuracy: 0.7506 -
loss: 0.7109 - val_accuracy: 0.6951 - val_loss: 0.8699
Epoch 8/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 81s 46ms/step - accuracy: 0.7701 -
loss: 0.6604 - val_accuracy: 0.7013 - val_loss: 0.8717
Epoch 9/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 70s 45ms/step - accuracy: 0.7850 -
loss: 0.6135 - val_accuracy: 0.7078 - val_loss: 0.8699
Epoch 10/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 84s 46ms/step - accuracy: 0.7956 -
loss: 0.5810 - val_accuracy: 0.7161 - val_loss: 0.8658
313/313 - 5s - 16ms/step - accuracy: 0.7161 - loss: 0.8658
0.7160999774932861

#-----------------CONFUSION MAATRIX--------------------------

y_pred = model.predict(test_images)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = test_labels.flatten()
cm = confusion_matrix(y_true, y_pred_classes)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.title("Confusion Matrix OF the CNN Modal")
plt.show()

313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 12ms/step
```



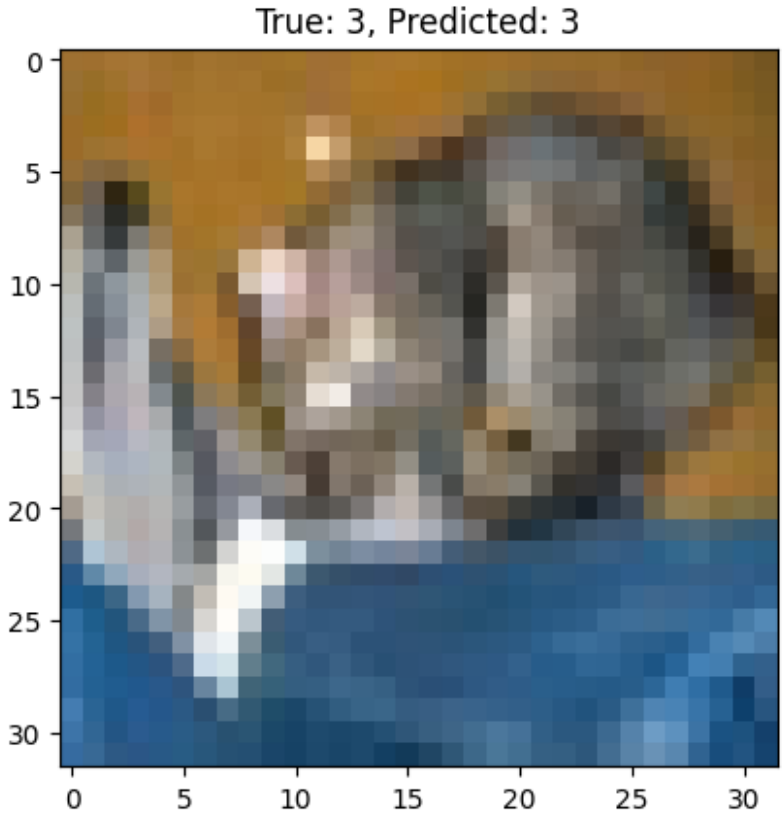

```
#----------------------ACCURACY
PLOT-------------------------------------------------------------

predictions = model.predict(test_images)
y_pred_classes = np.argmax(y_pred, axis=1)
y_true = test_labels.flatten()
cm = confusion_matrix(y_true, y_pred_classes)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.title("Confusion Matrix")
plt.show()

predicted_classes = np.argmax(predictions, axis=1)
true_classes = test_labels.flatten()


print(classification_report(y_true, y_pred_classes)) #Correction
Report

# Show first 5 test images and predictions
for i in range(100):
    plt.imshow(test_images[i])
    plt.title(f"True: {true_classes[i]}, Predicted:
```

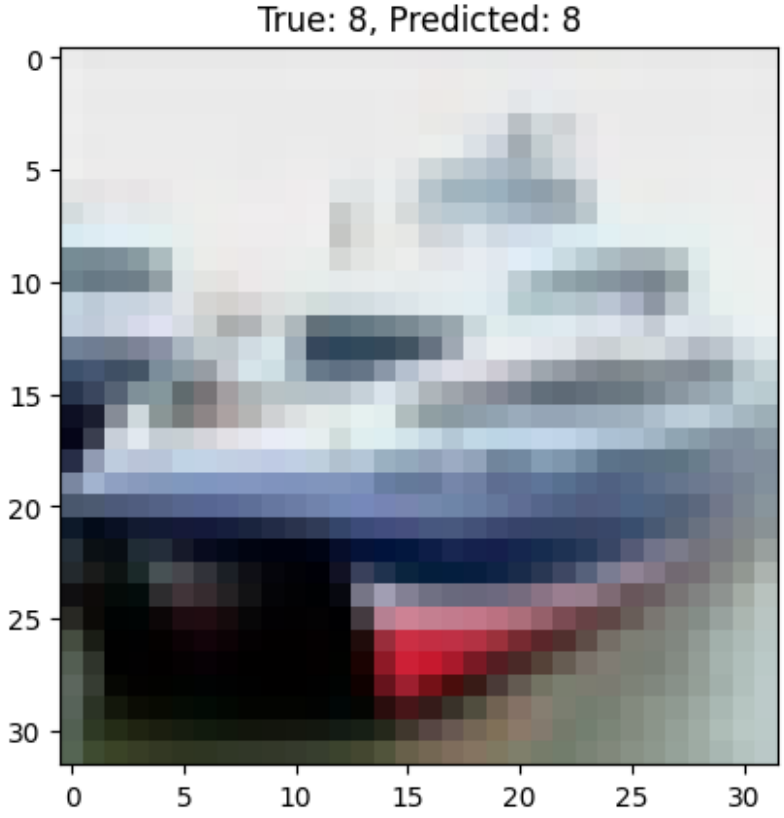
```
{predicted_classes[i]}")
    plt.show()
```

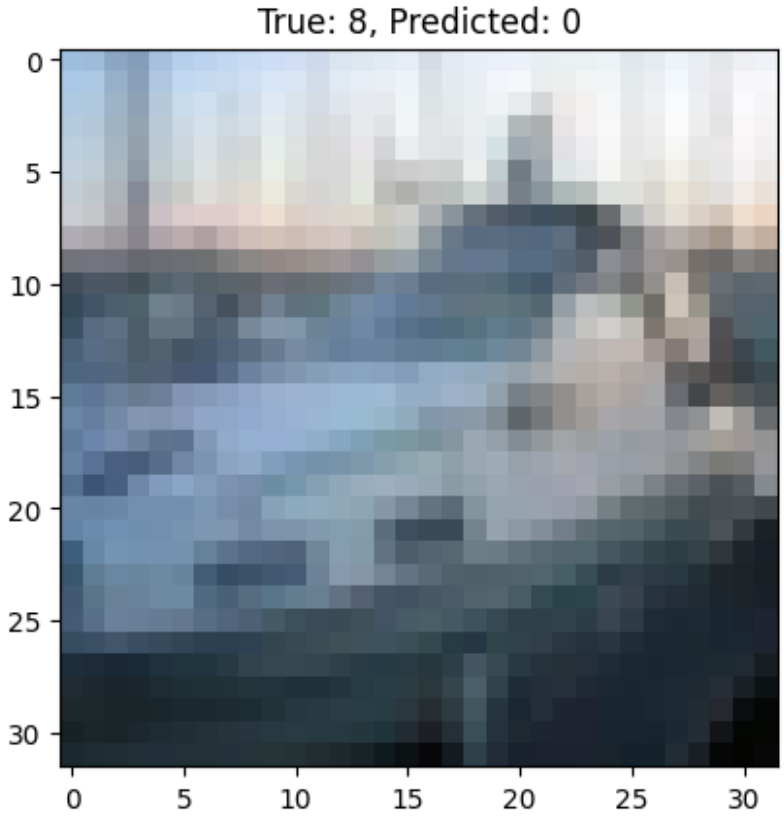
```
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 13ms/step
```

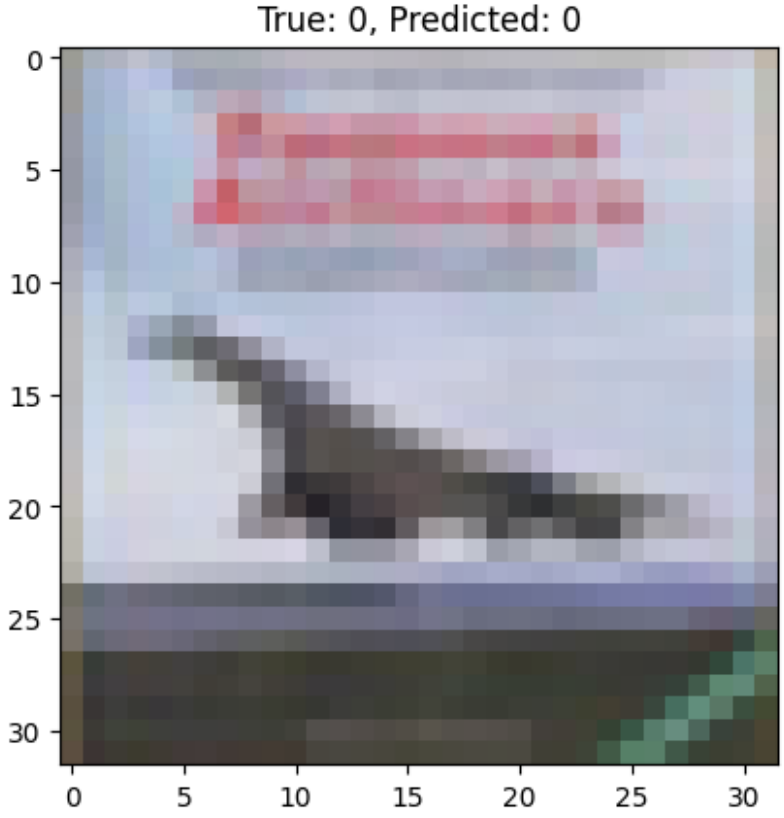


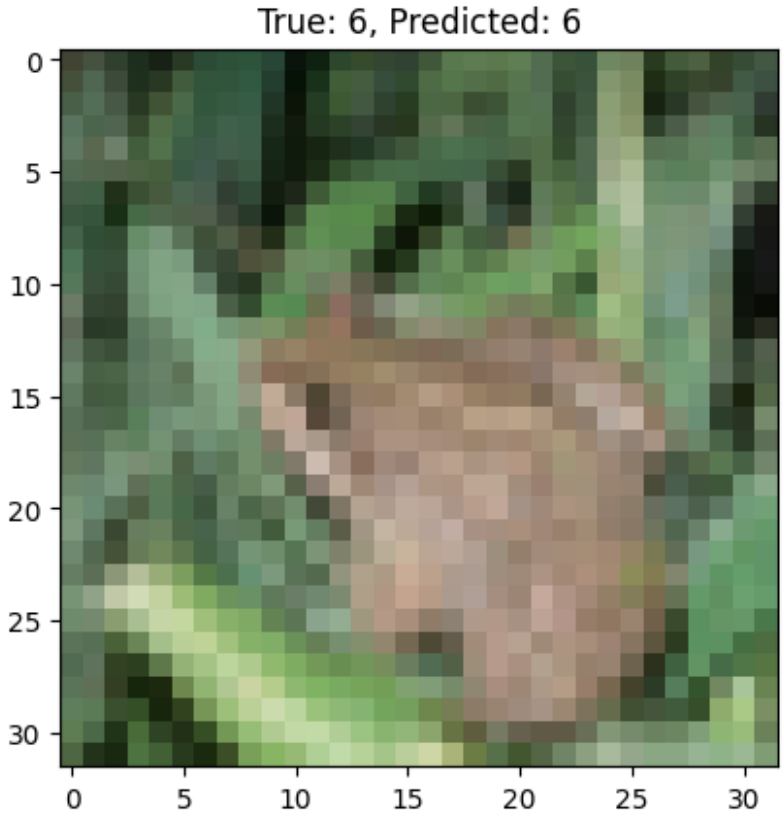
	precision	recall	f1-score	support
0	0.75	0.76	0.75	1000
1	0.84	0.83	0.84	1000
2	0.65	0.57	0.61	1000
3	0.60	0.45	0.51	1000
4	0.62	0.74	0.68	1000
5	0.60	0.63	0.61	1000
6	0.71	0.81	0.76	1000
7	0.75	0.75	0.75	1000
8	0.80	0.84	0.82	1000
9	0.84	0.77	0.80	1000
accuracy			0.72	10000
macro avg	0.72	0.72	0.71	10000
weighted avg	0.72	0.72	0.71	10000

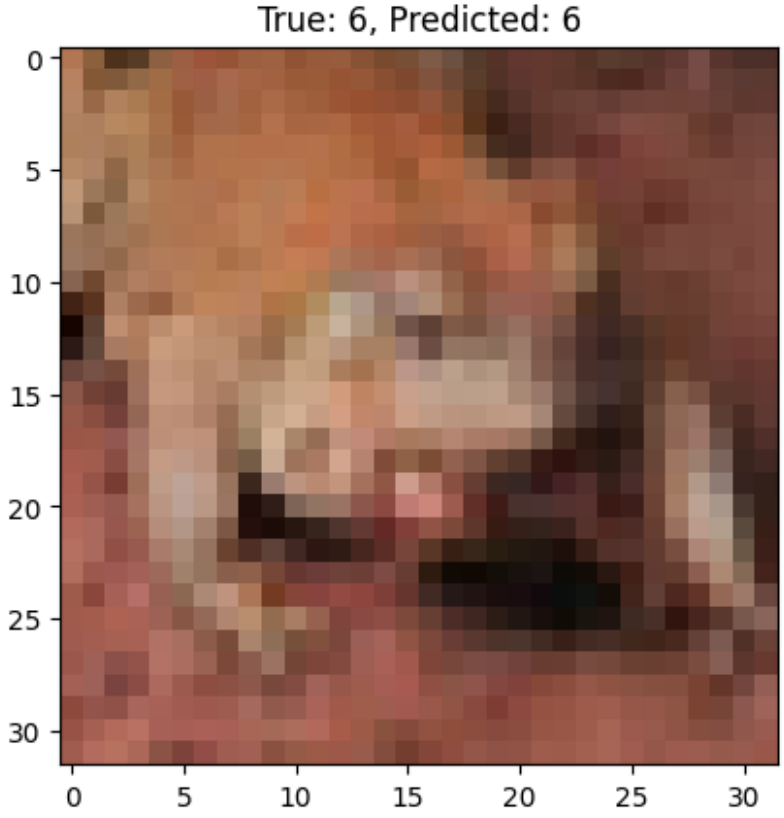

True: 3, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

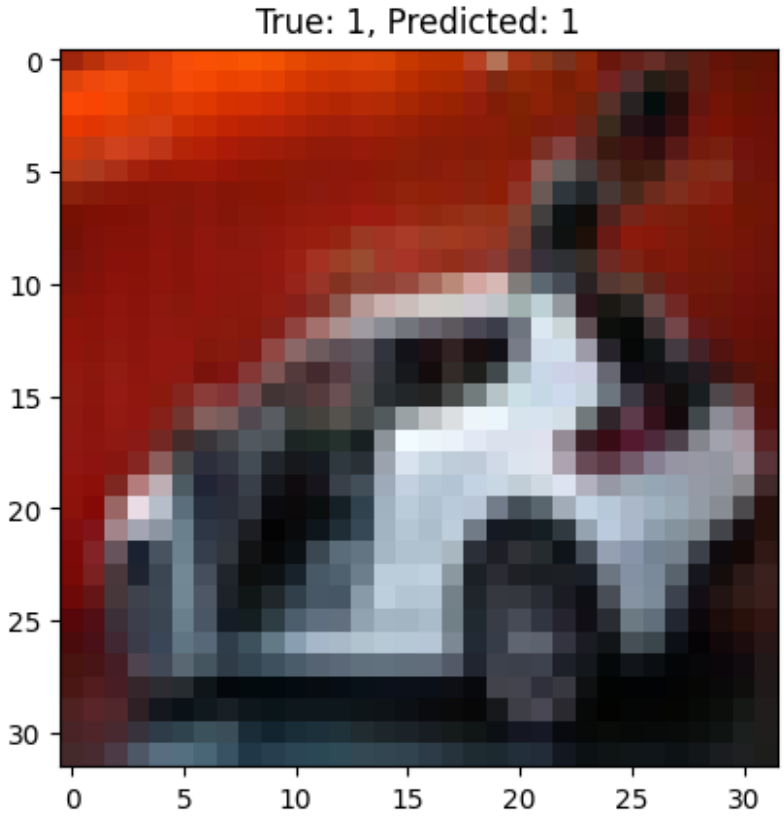

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

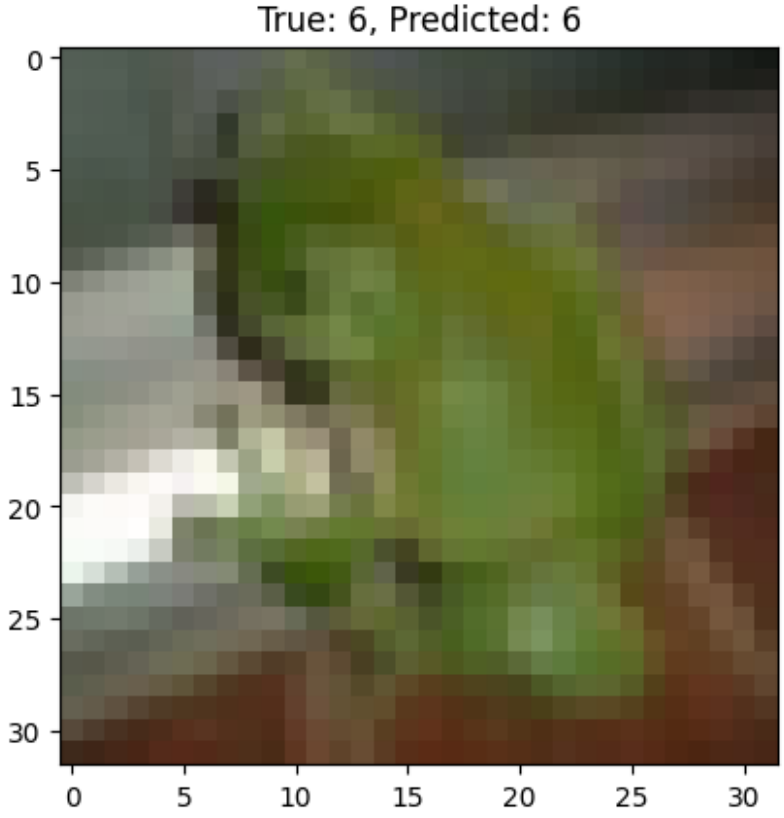

True: 8, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30

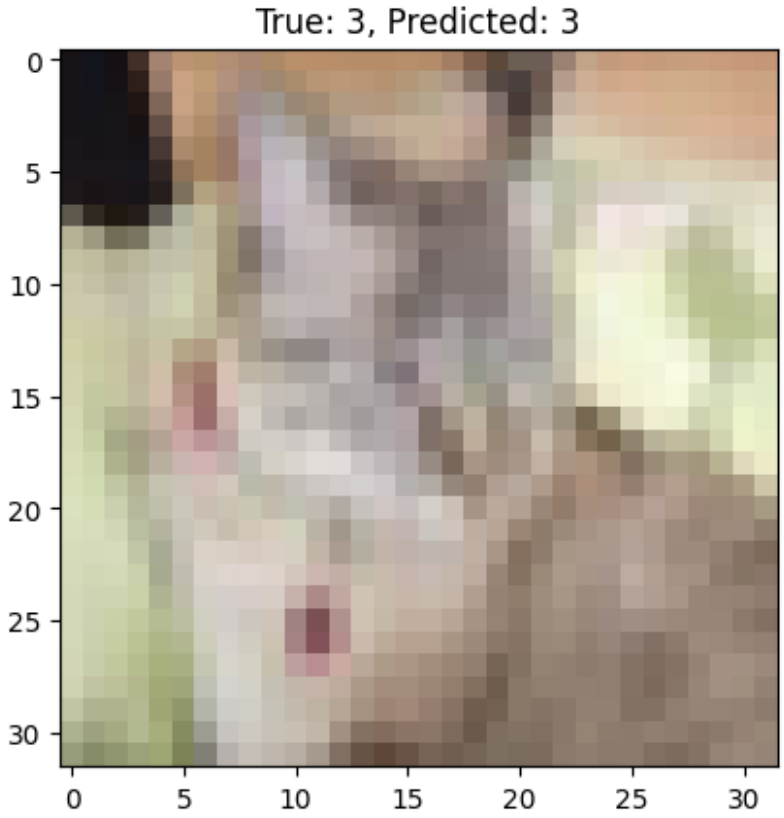

True: 0, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30

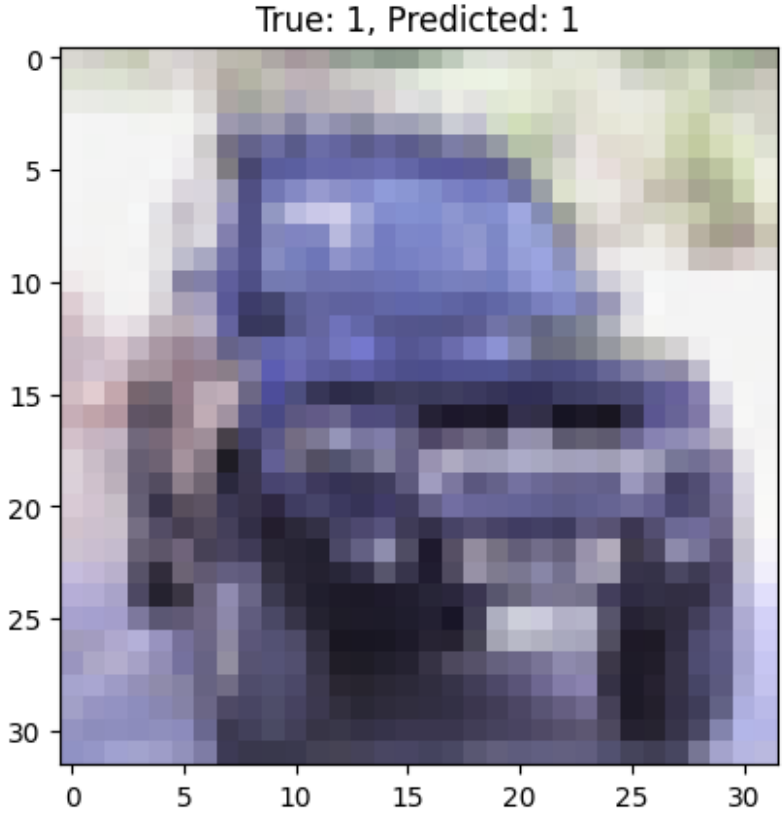

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

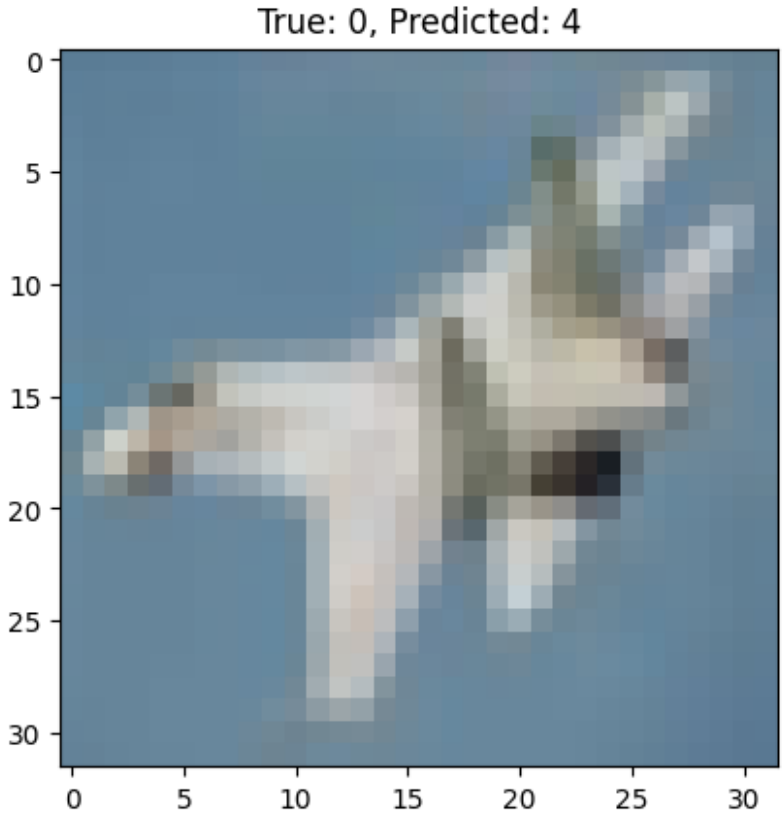

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

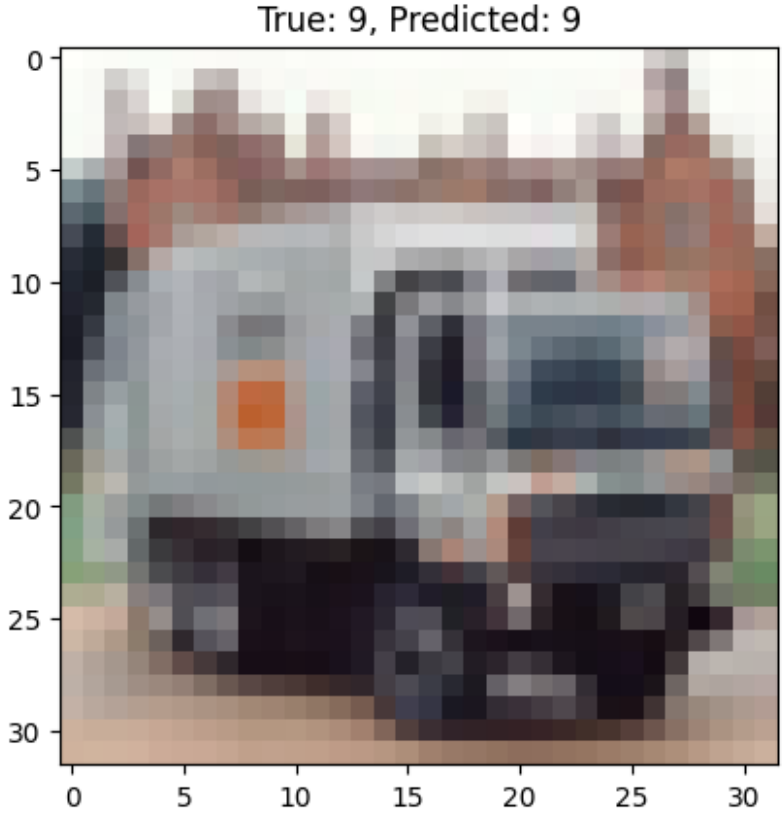

True: 1, Predicted: 1
0
5
10
15
20
25
30
0
5
10
15
20
25
30

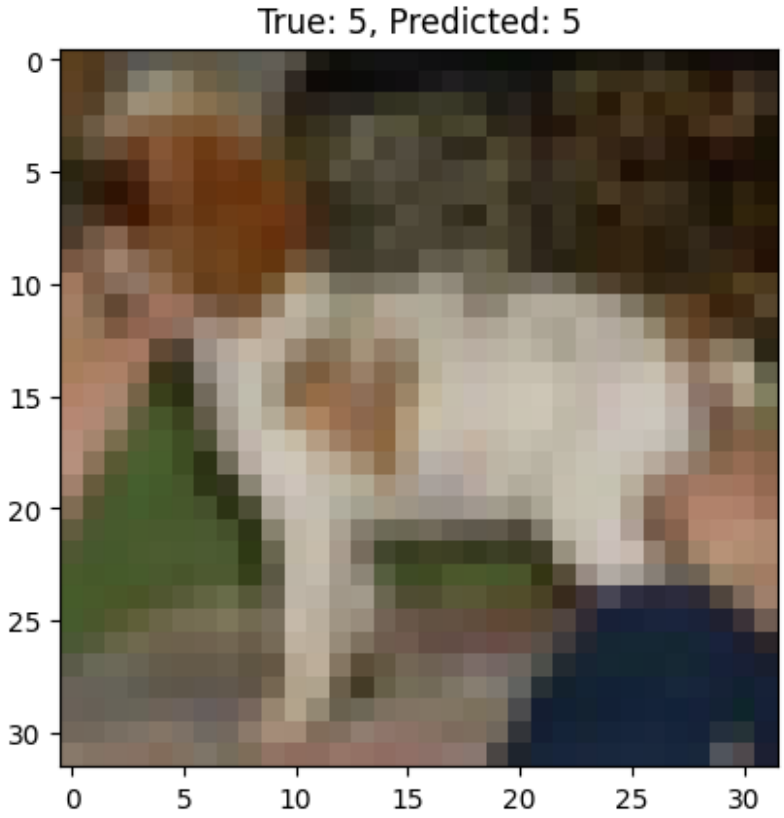

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

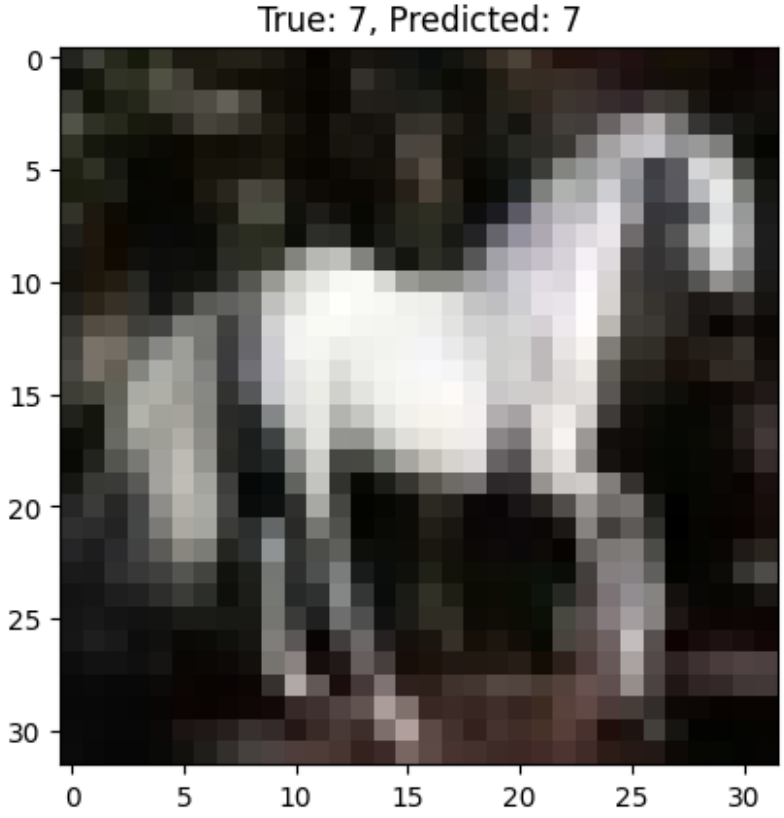

True: 3, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

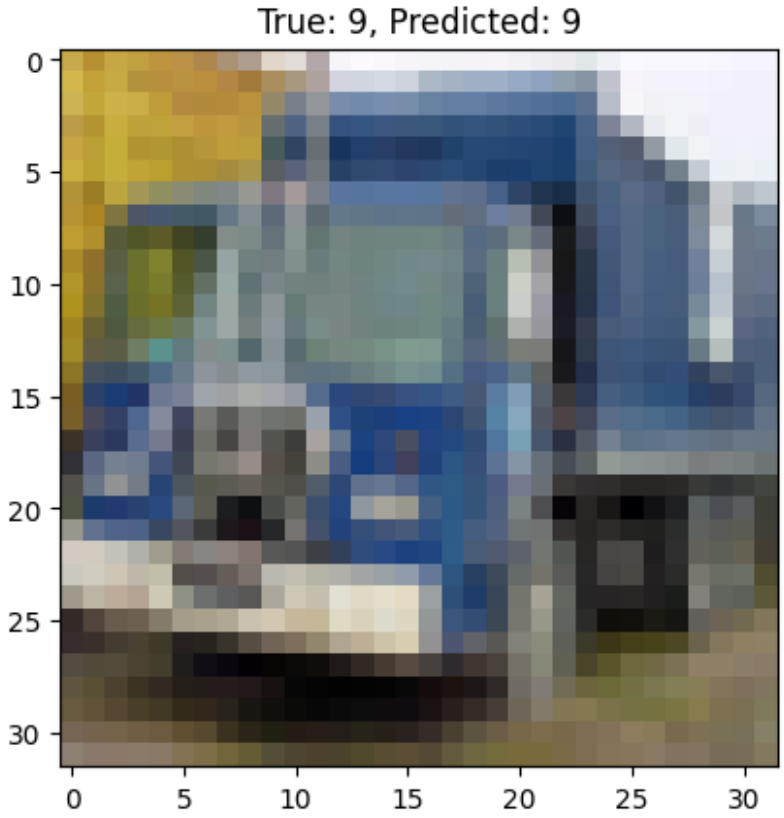

True: 1, Predicted: 1
0
5
10
15
20
25
30
0
5
10
15
20
25
30

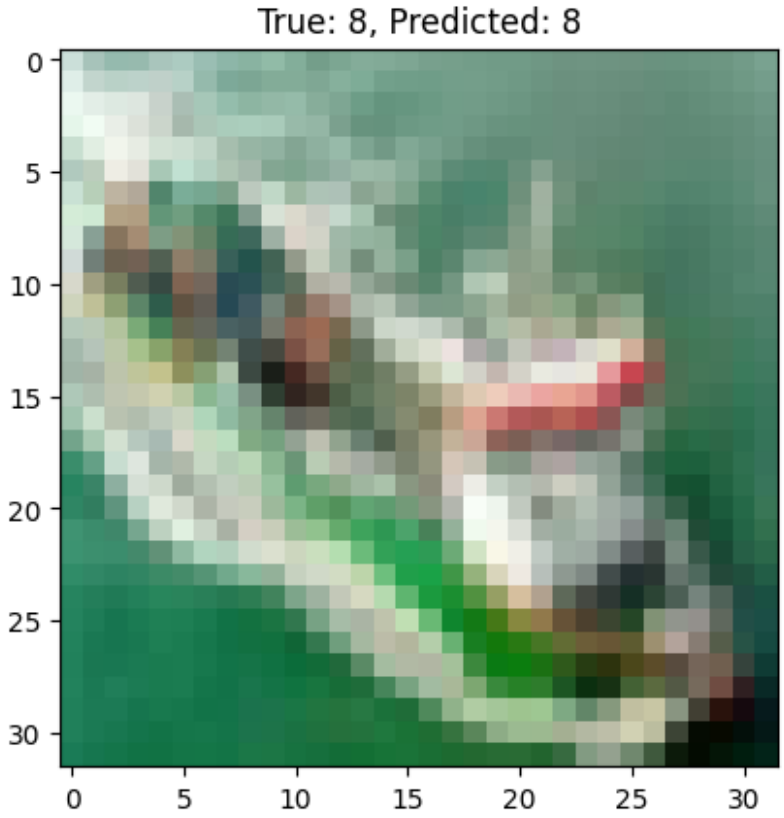

True: 0, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30

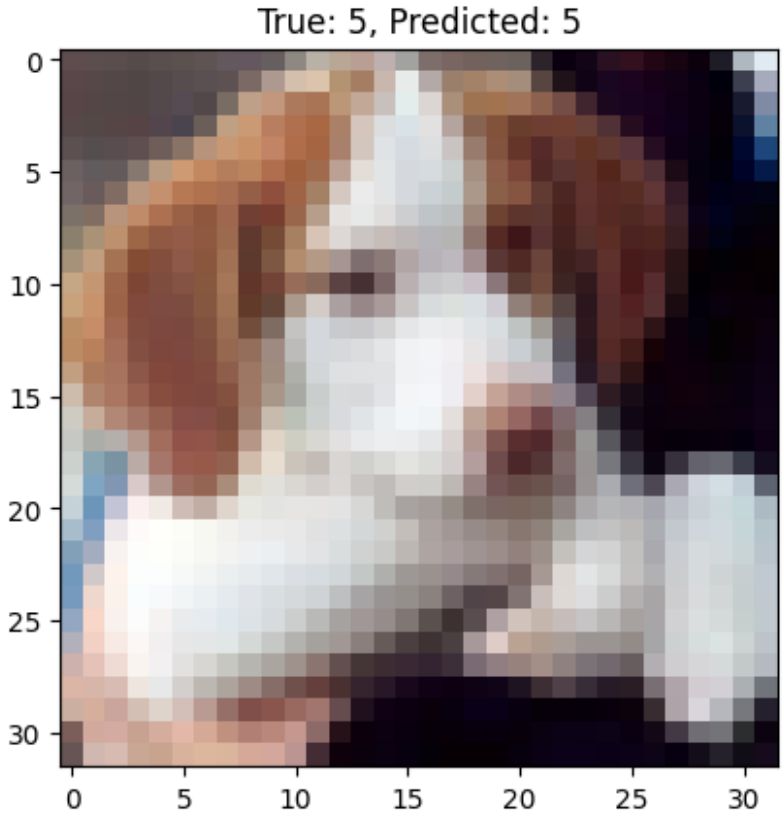

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

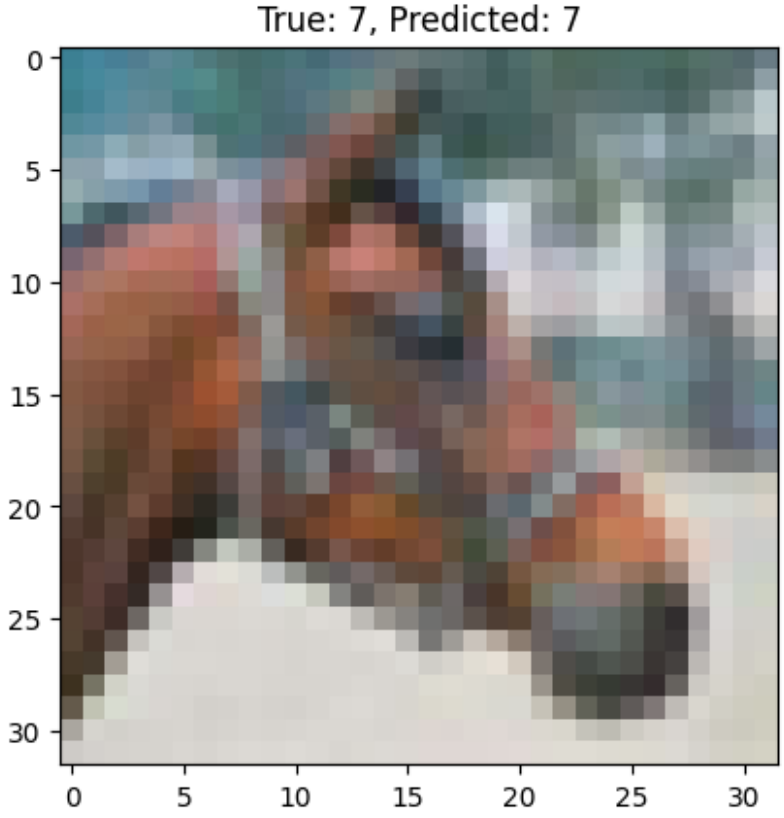

True: 5, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

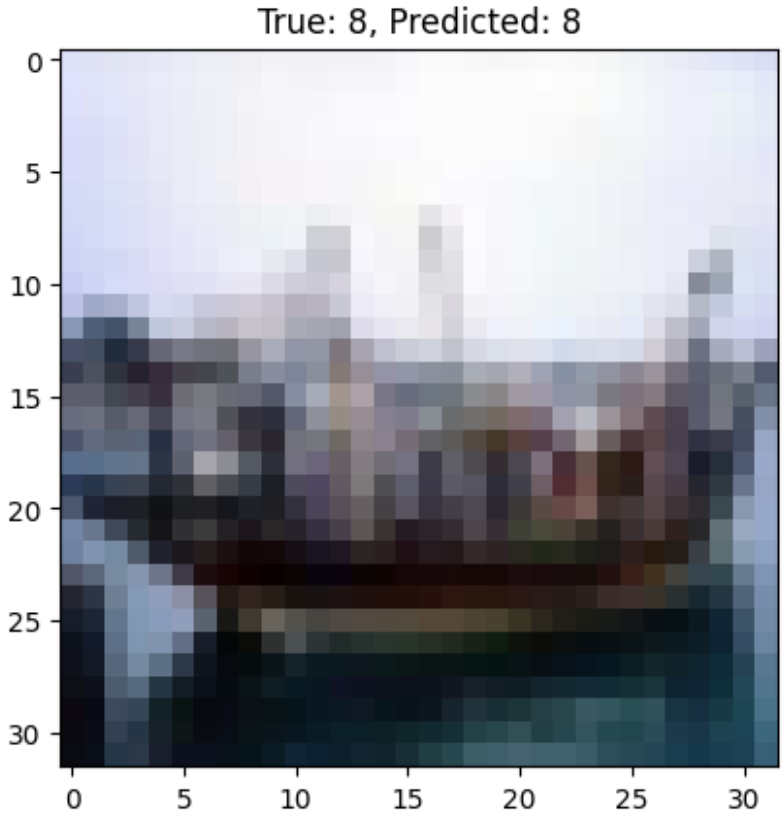

True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

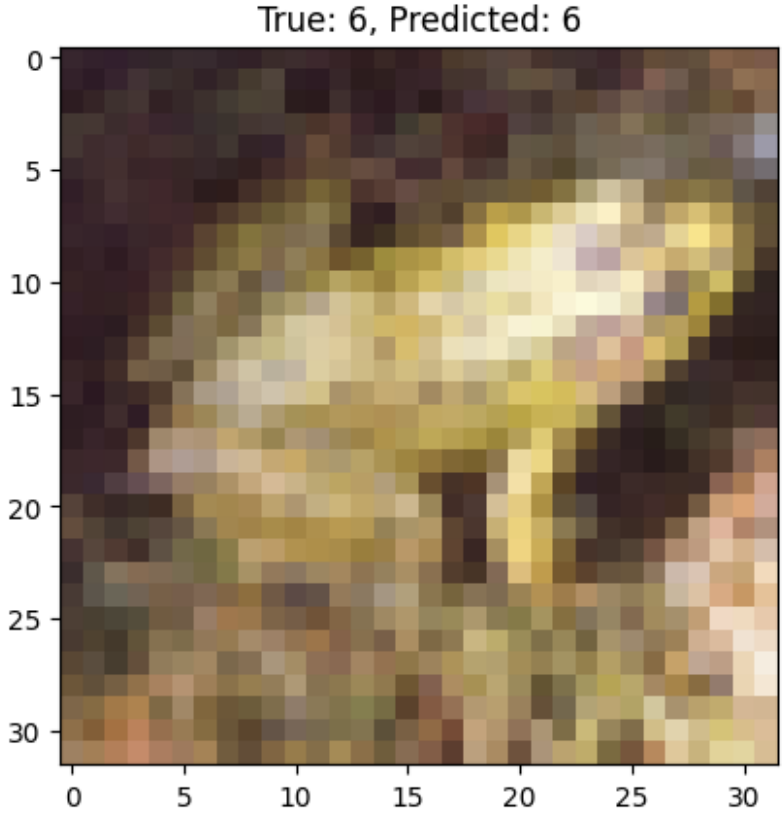

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

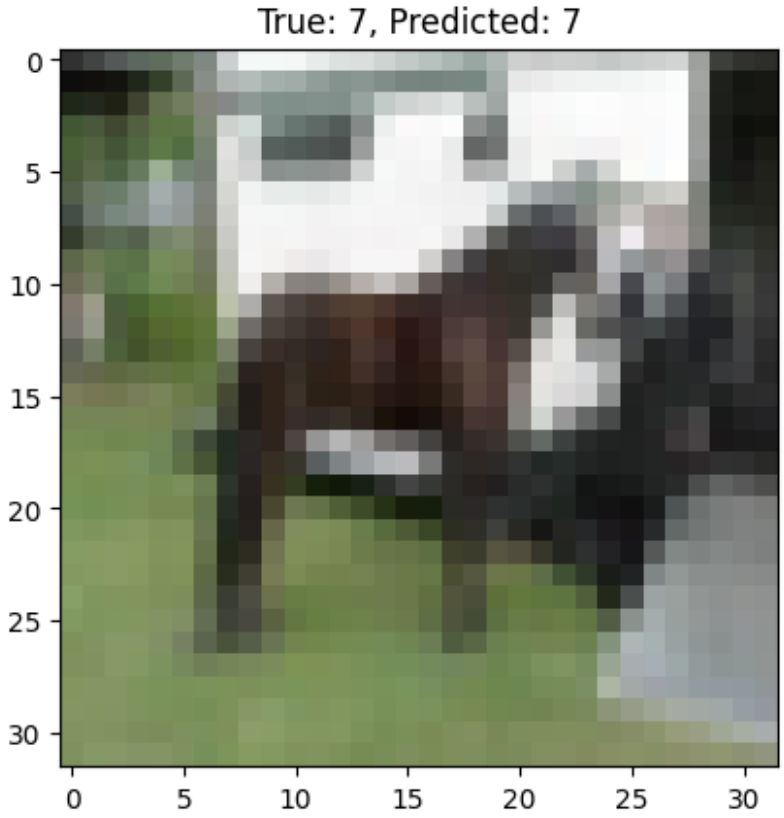

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

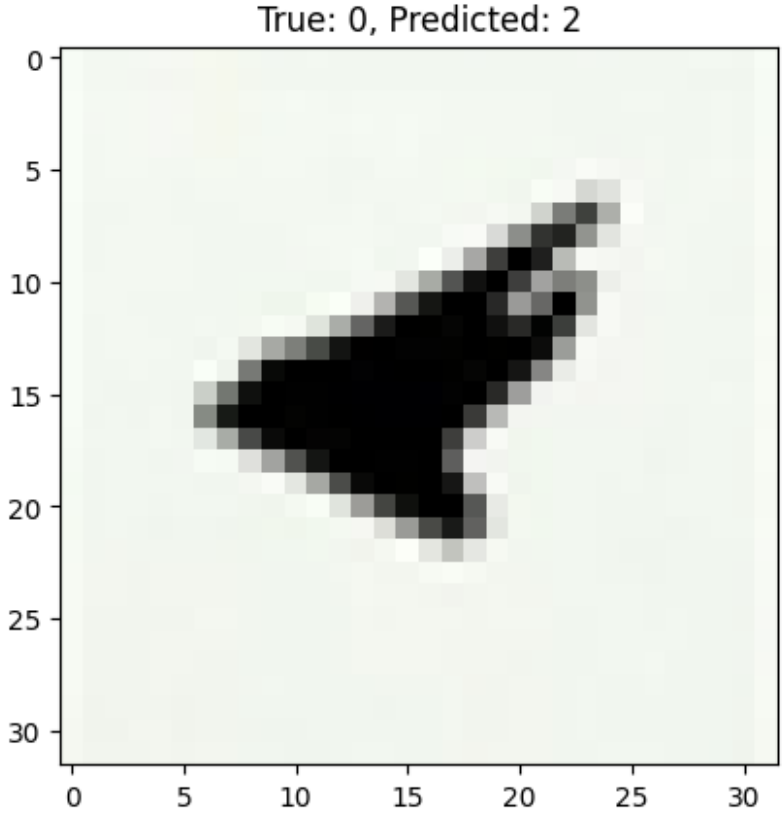

True: 5, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

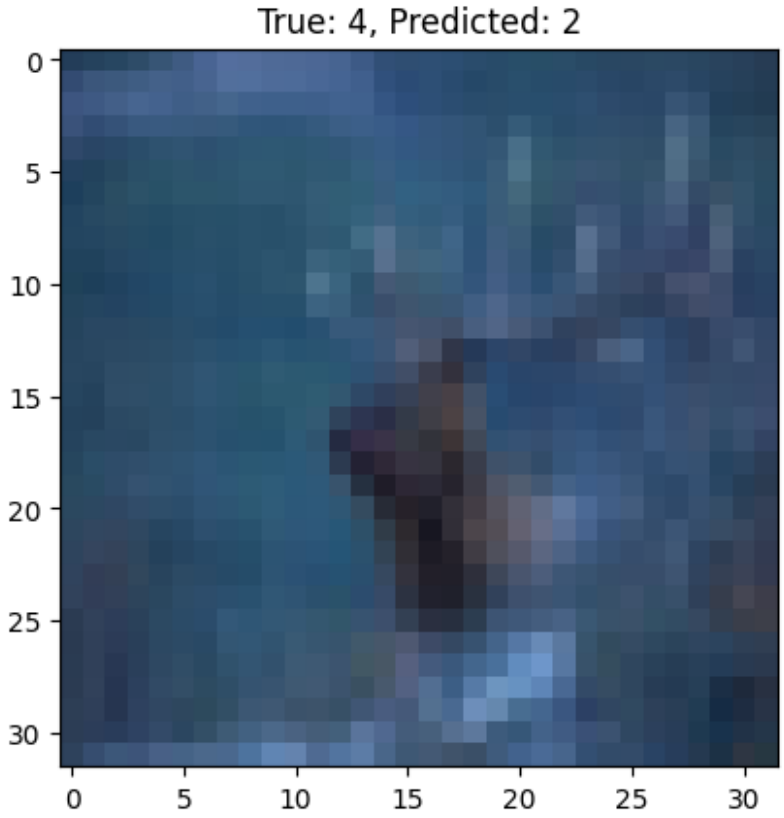

True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

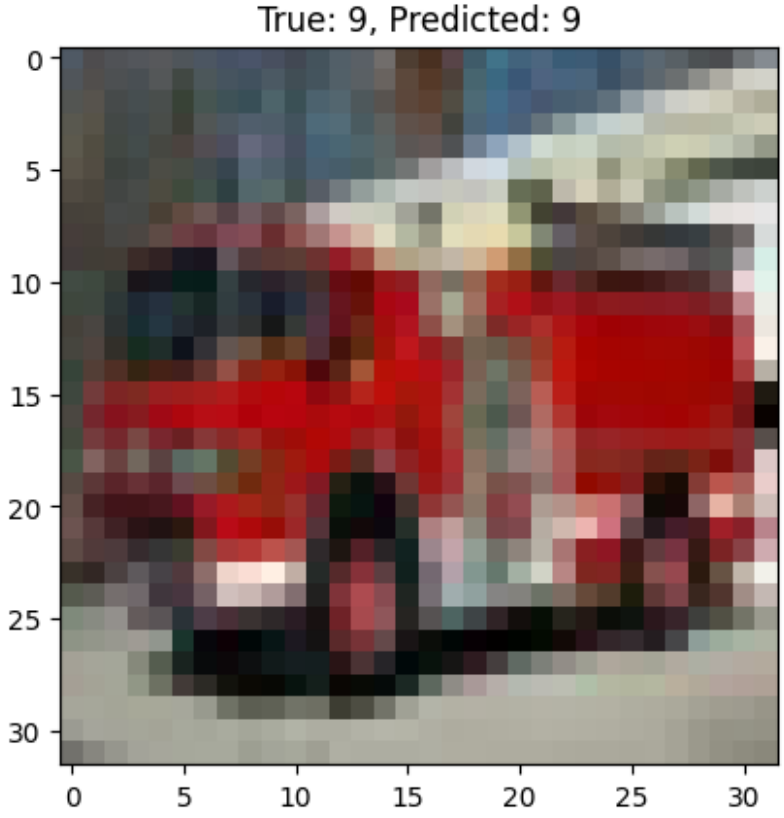

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

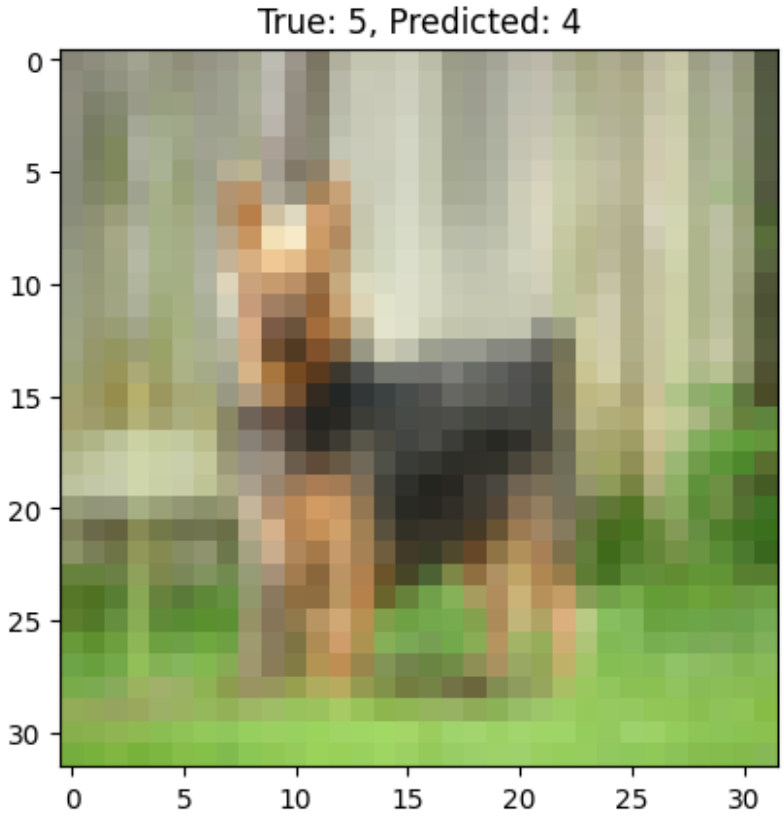

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

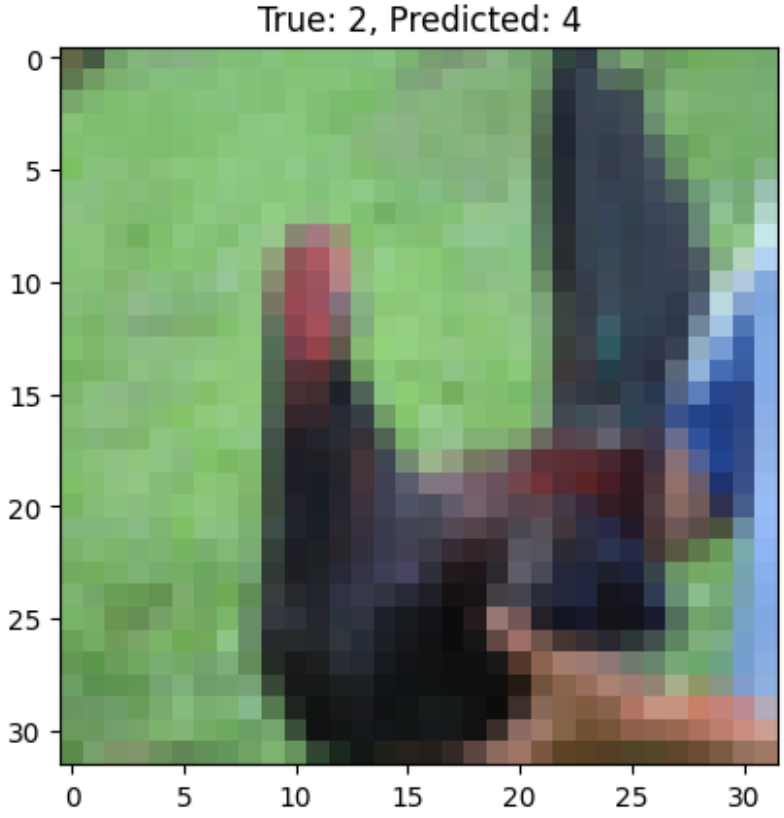

True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

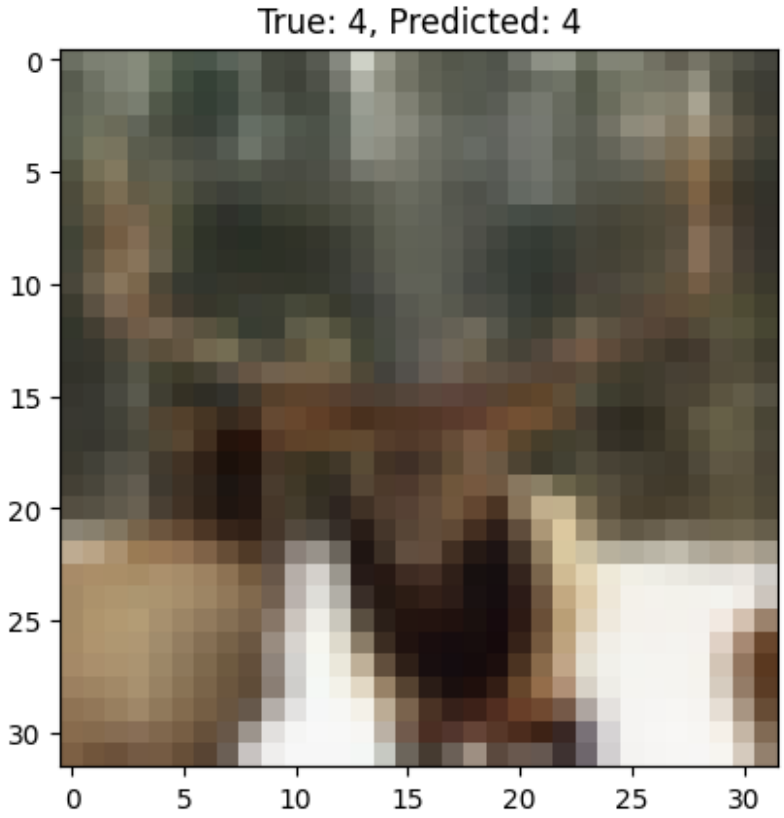

True: 0, Predicted: 2
0
5
10
15
20
25
30
0
5
10
15
20
25
30

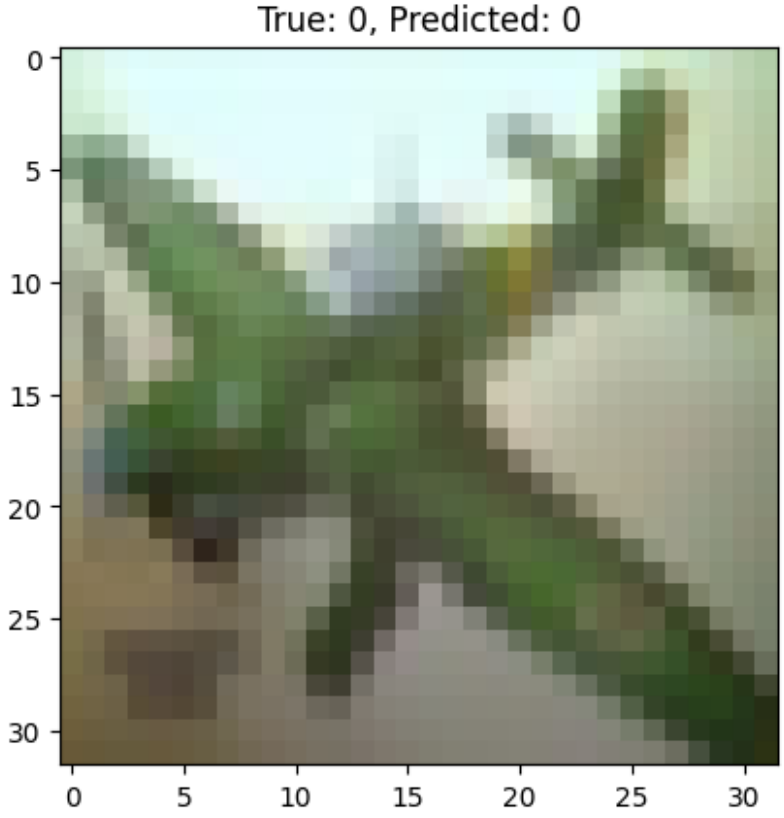

True: 4, Predicted: 2
0
5
10
15
20
25
30
0
5
10
15
20
25
30

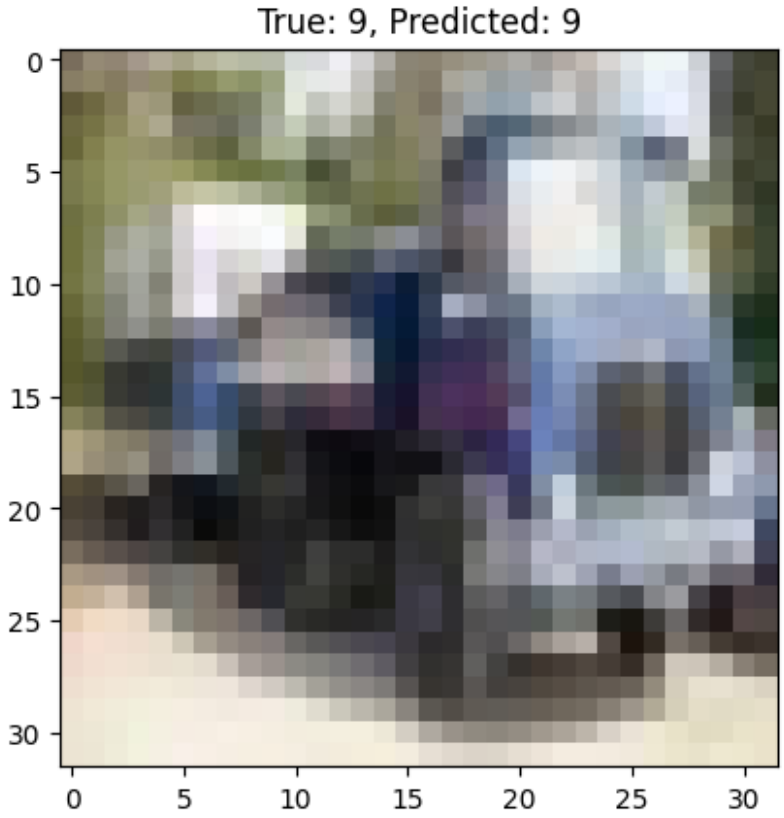

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

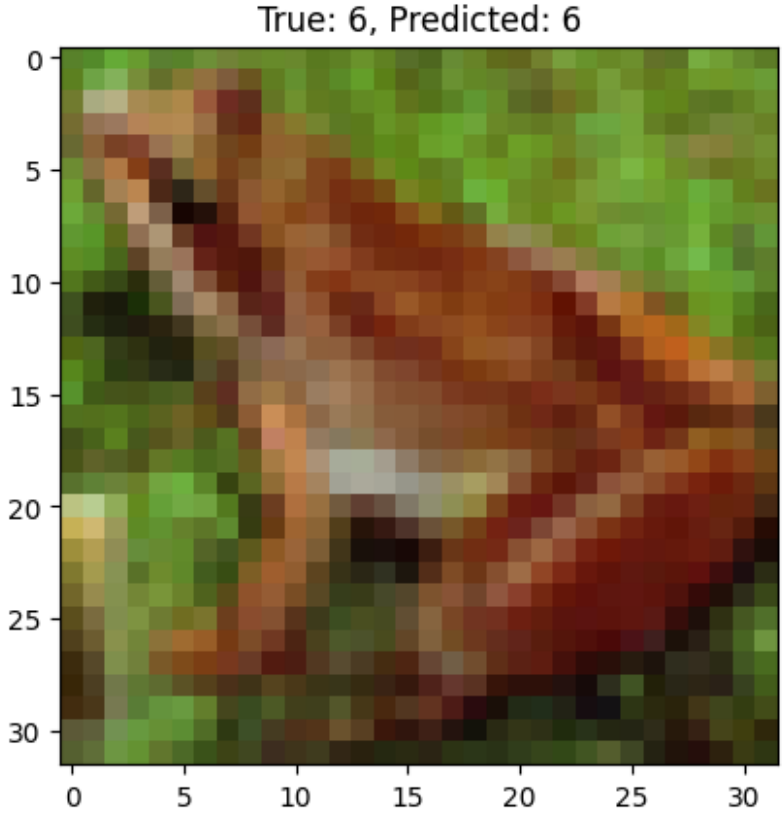

True: 5, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30

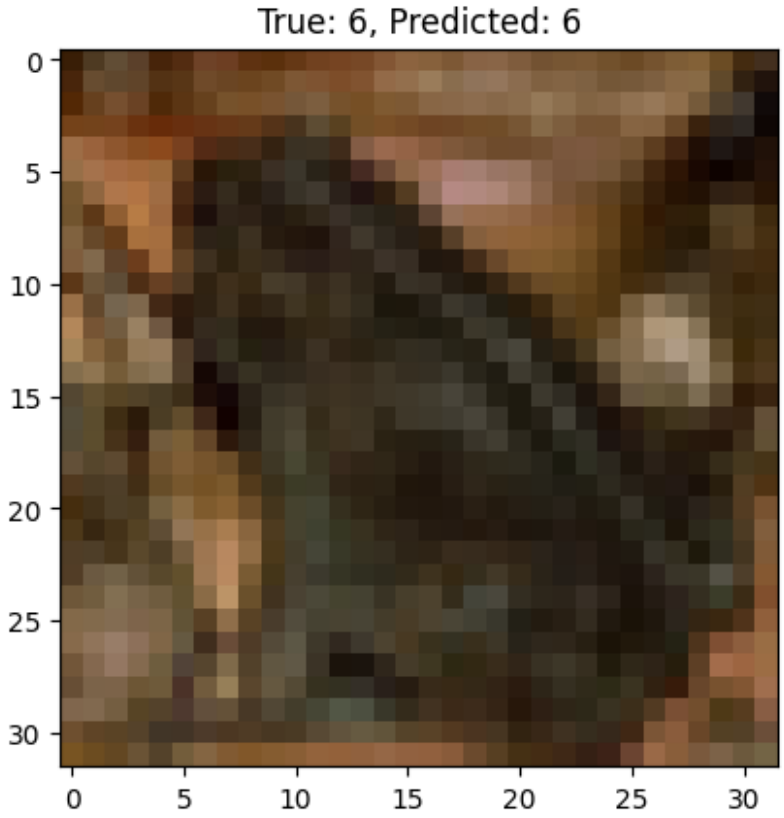

True: 2, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30

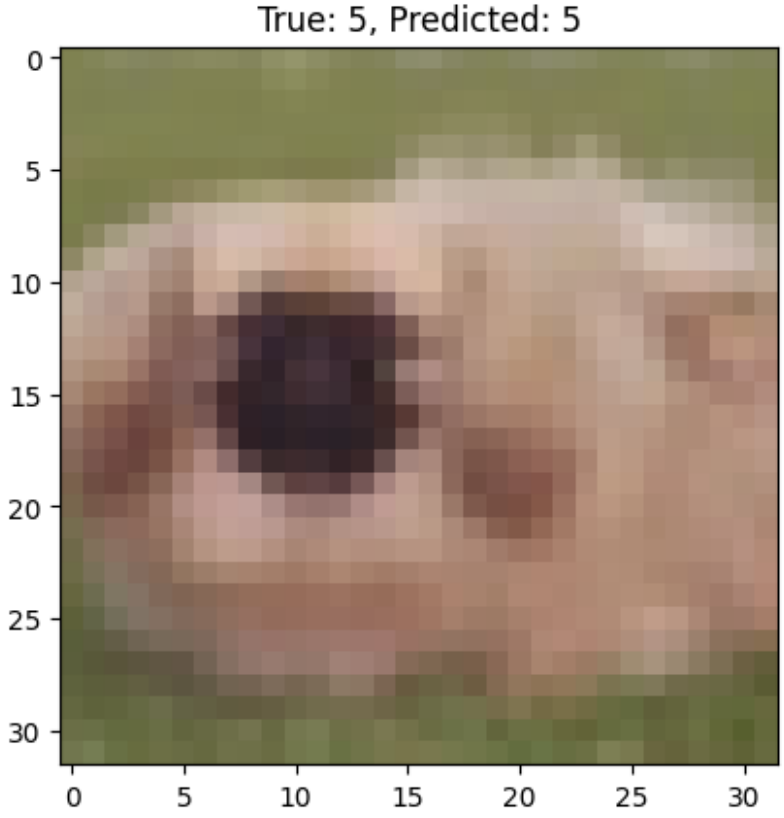

True: 4, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30

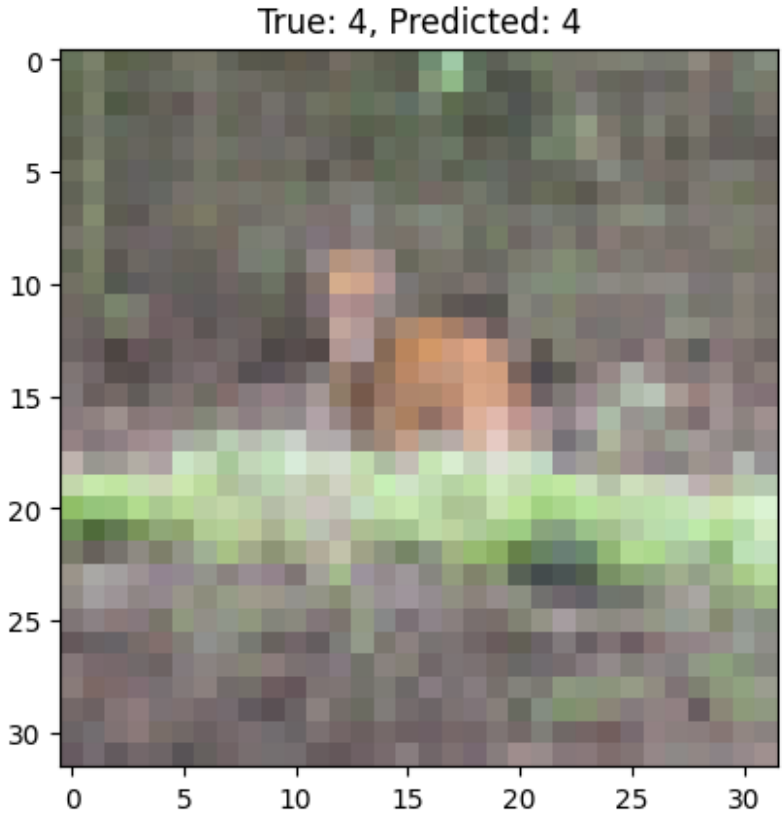

True: 0, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30

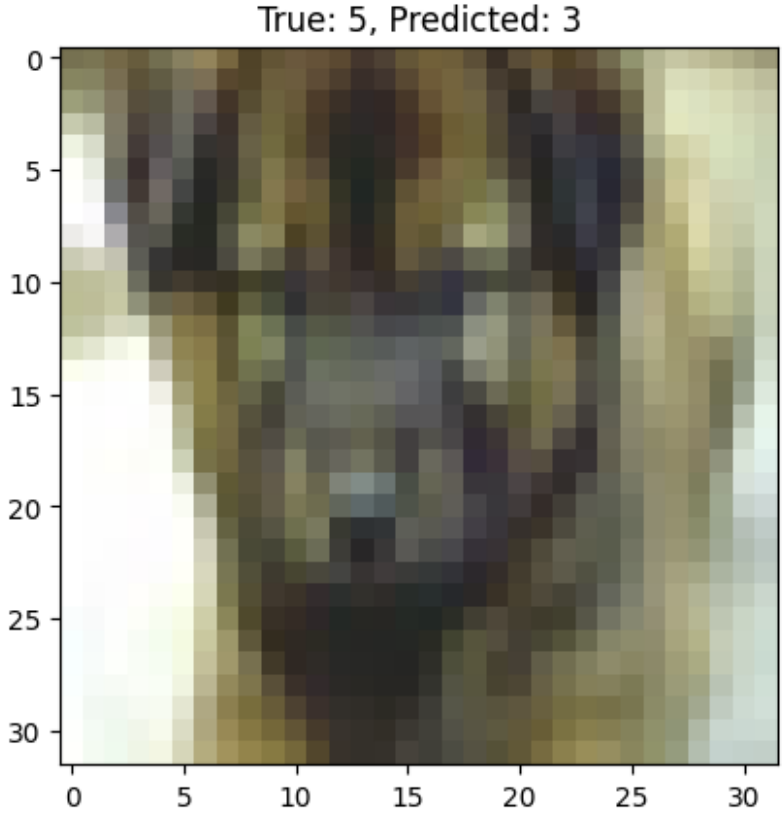

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

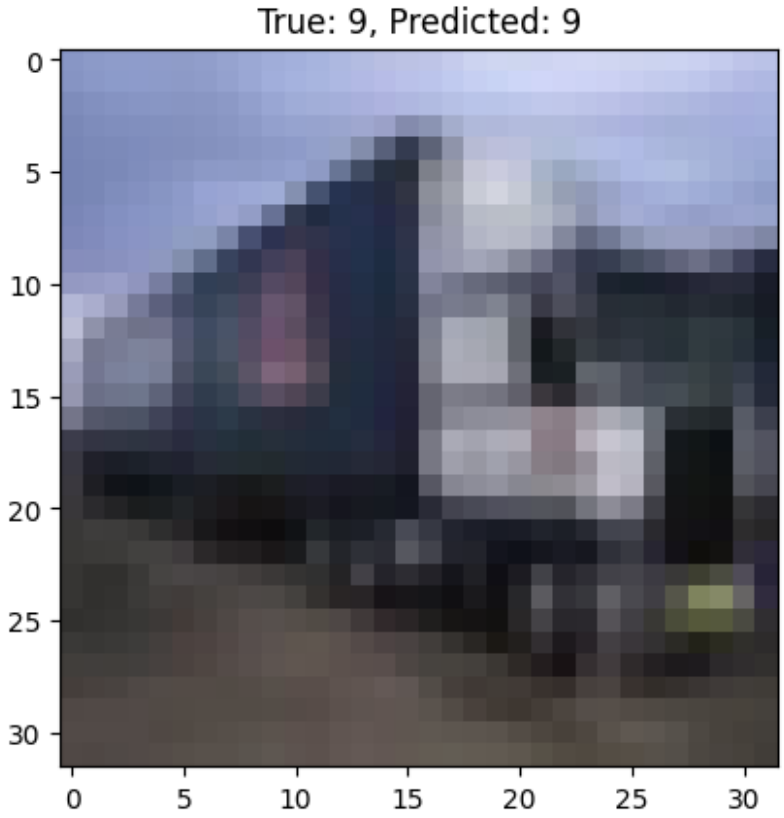

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

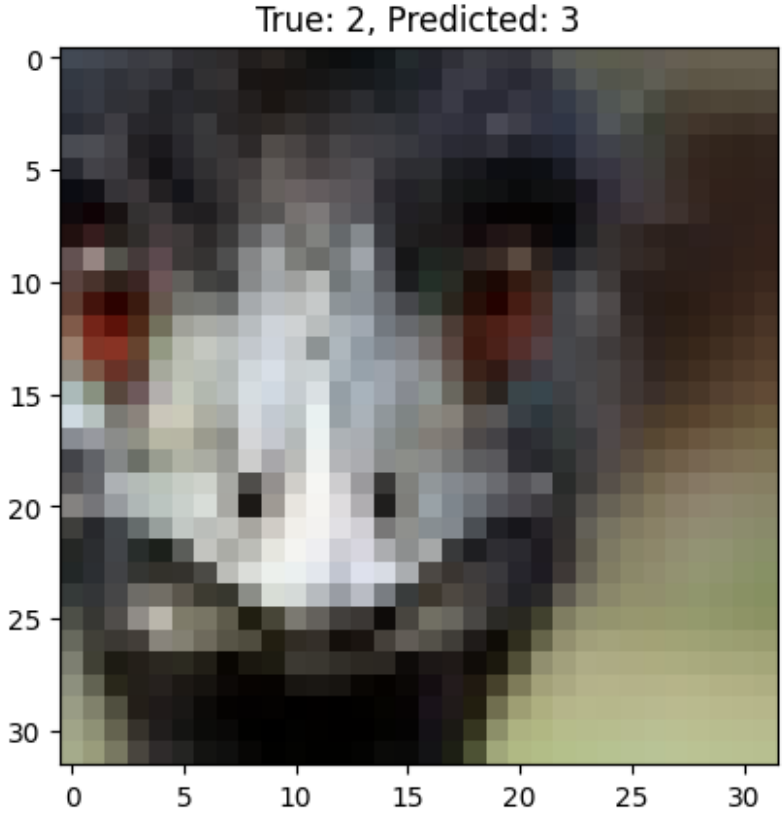

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

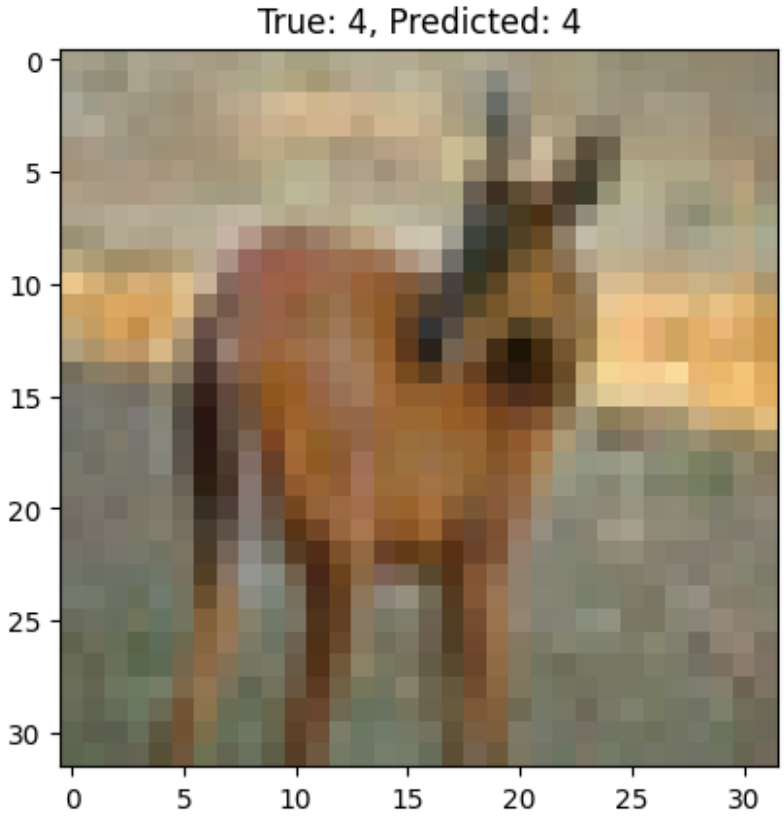

True: 5, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

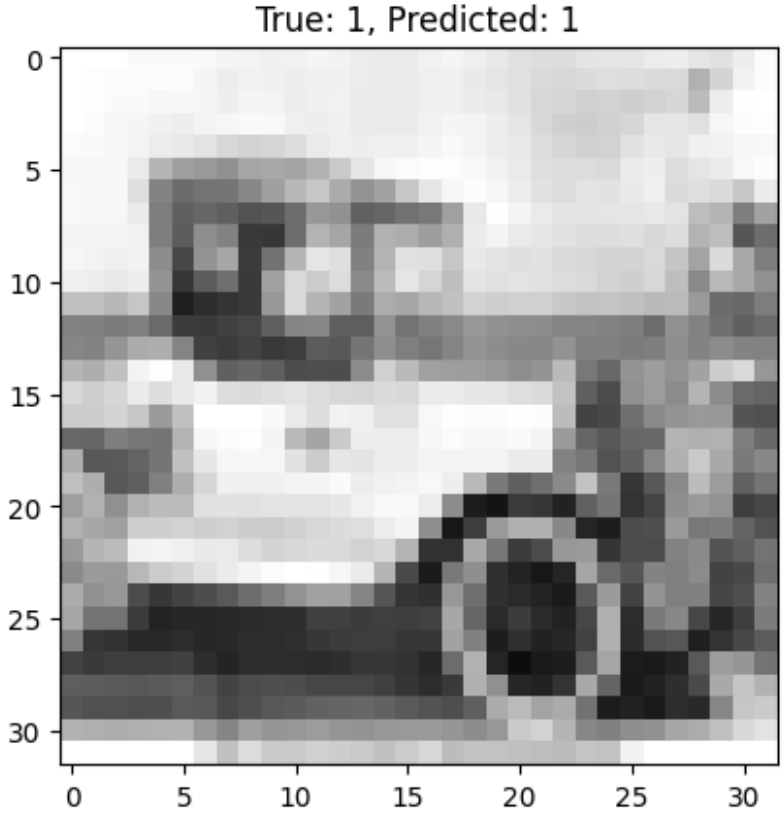

True: 4, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30

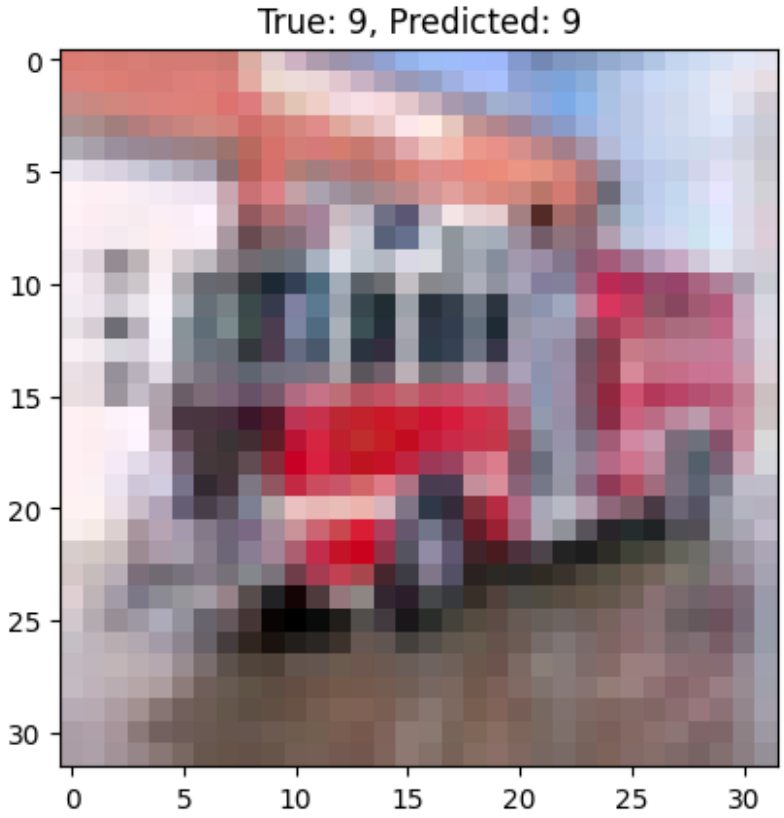

True: 5, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

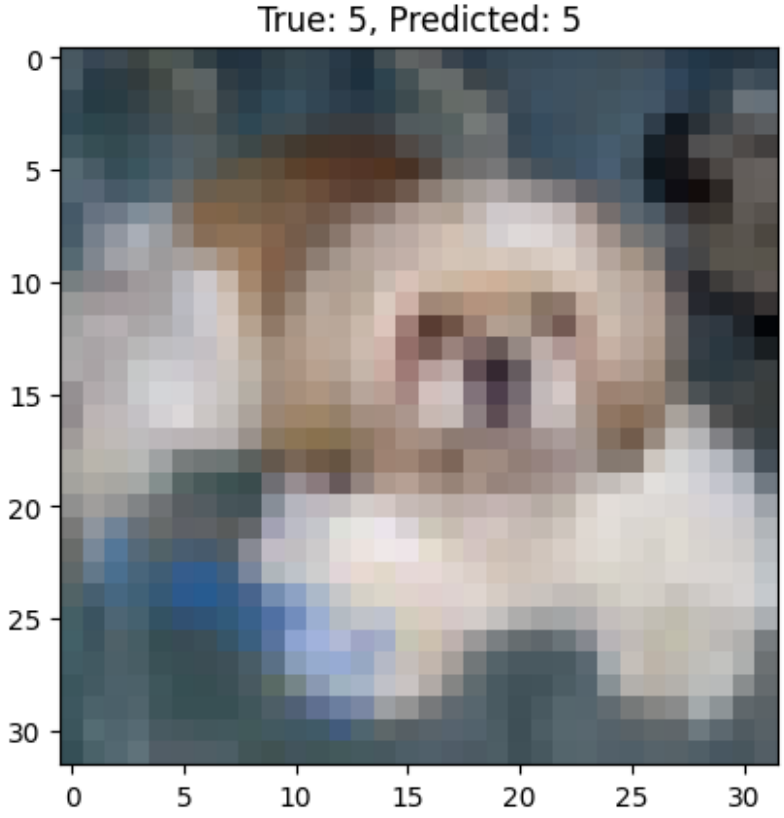

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

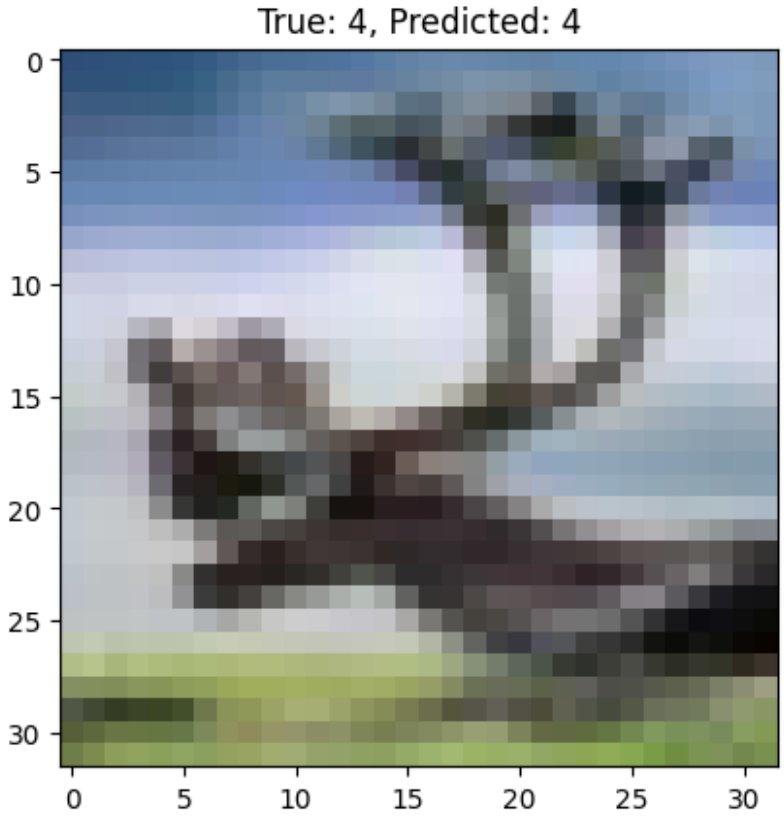

True: 2, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

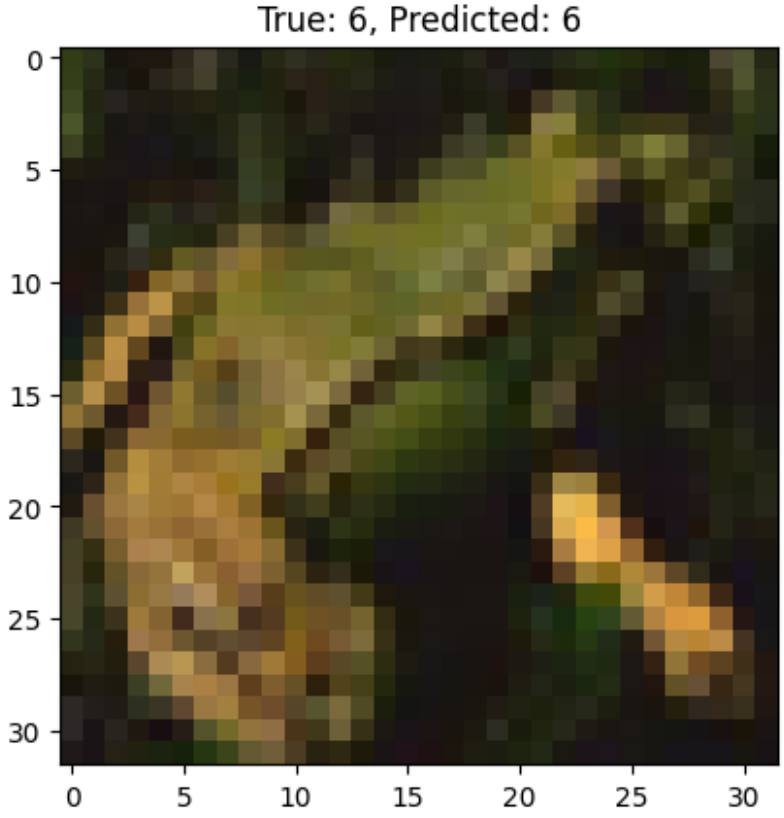

True: 4, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30

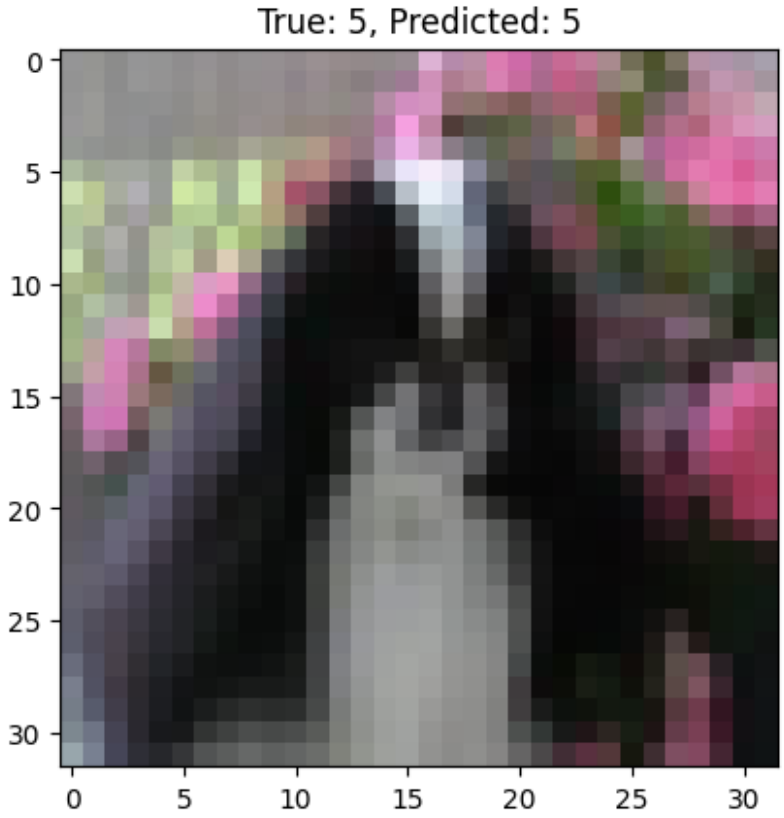

True: 1, Predicted: 1
0
5
10
15
20
25
30
0
5
10
15
20
25
30

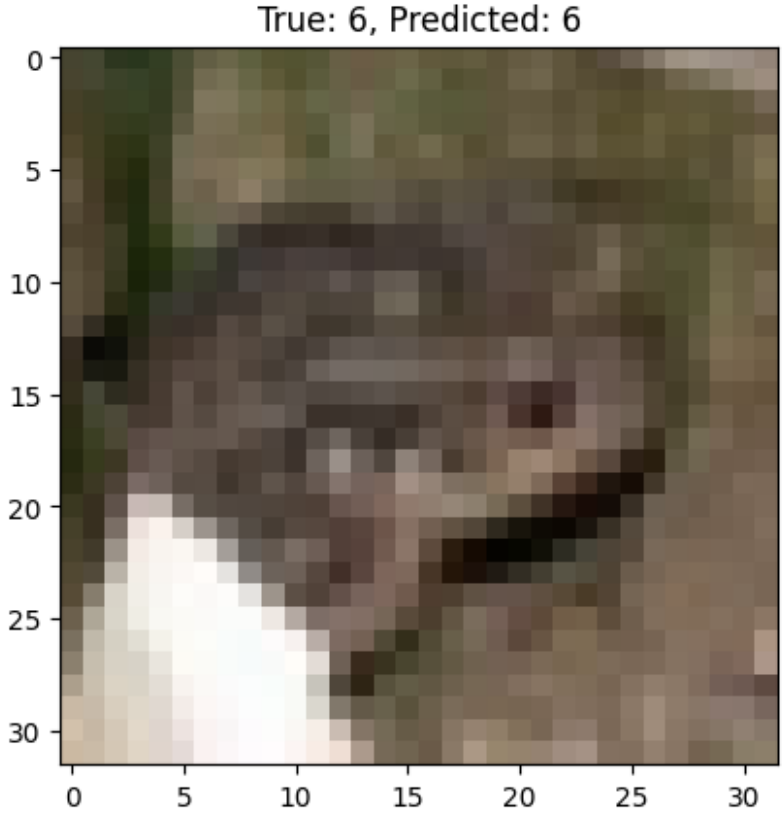

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

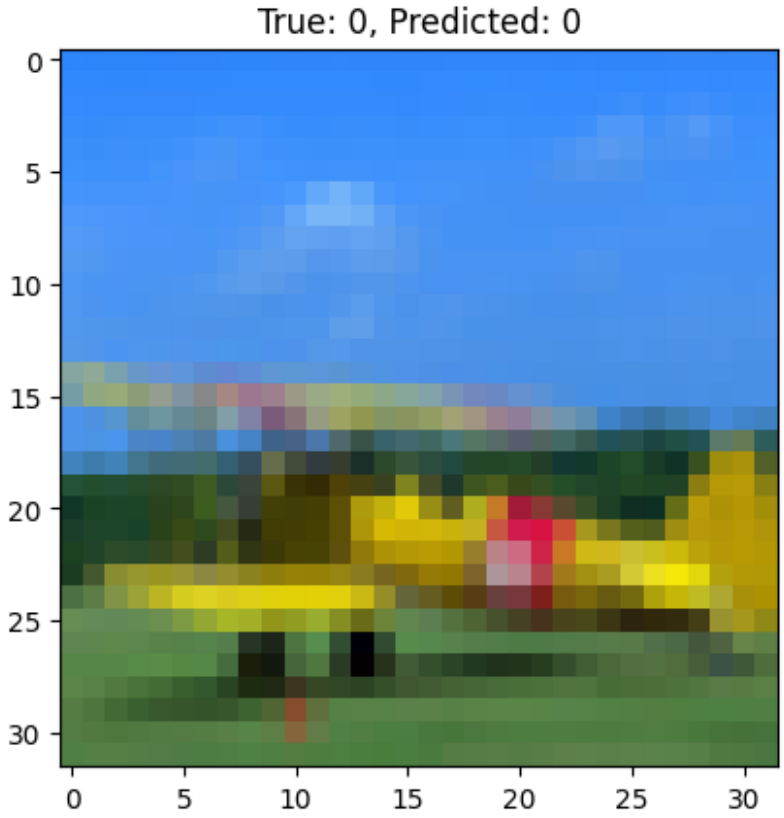

True: 5, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

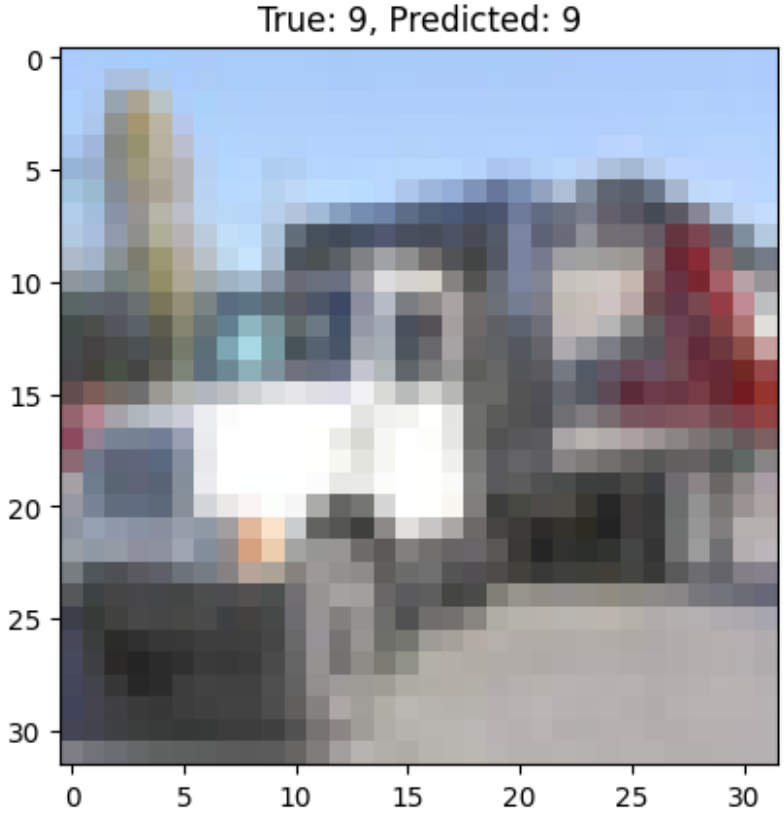

True: 4, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30

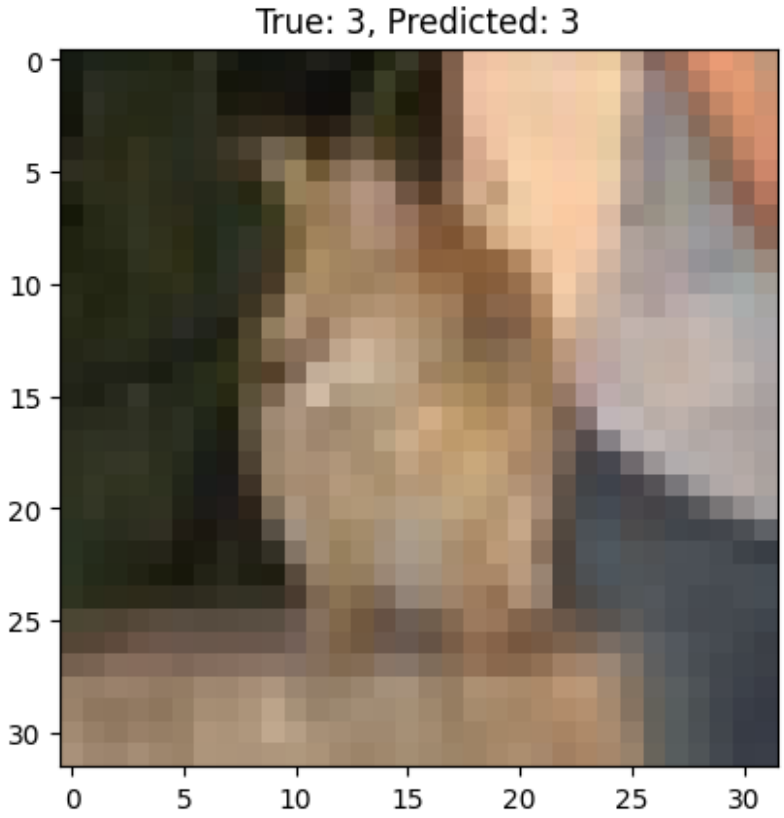

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

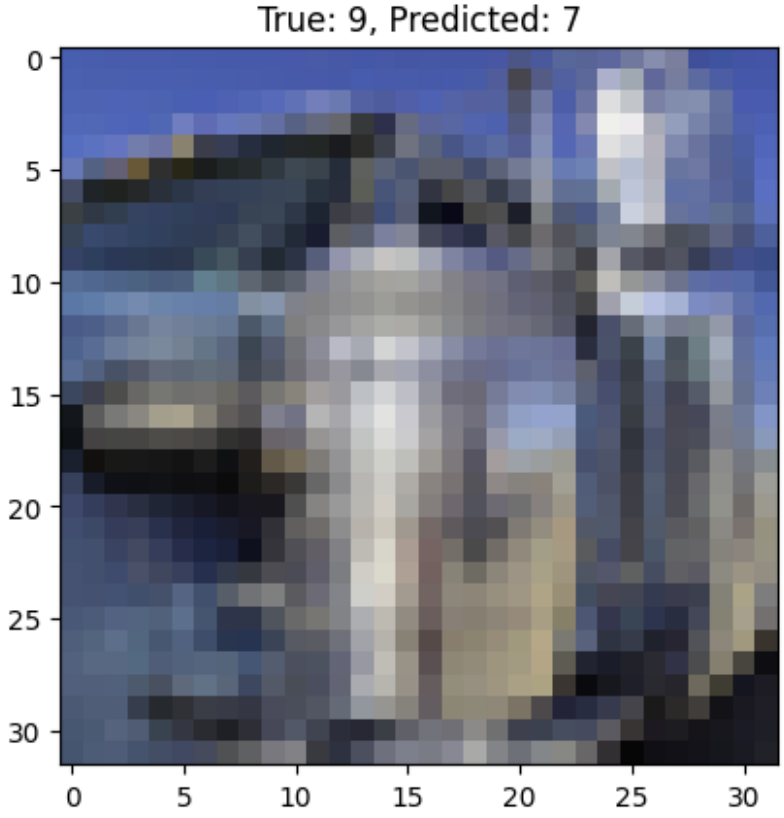

True: 5, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

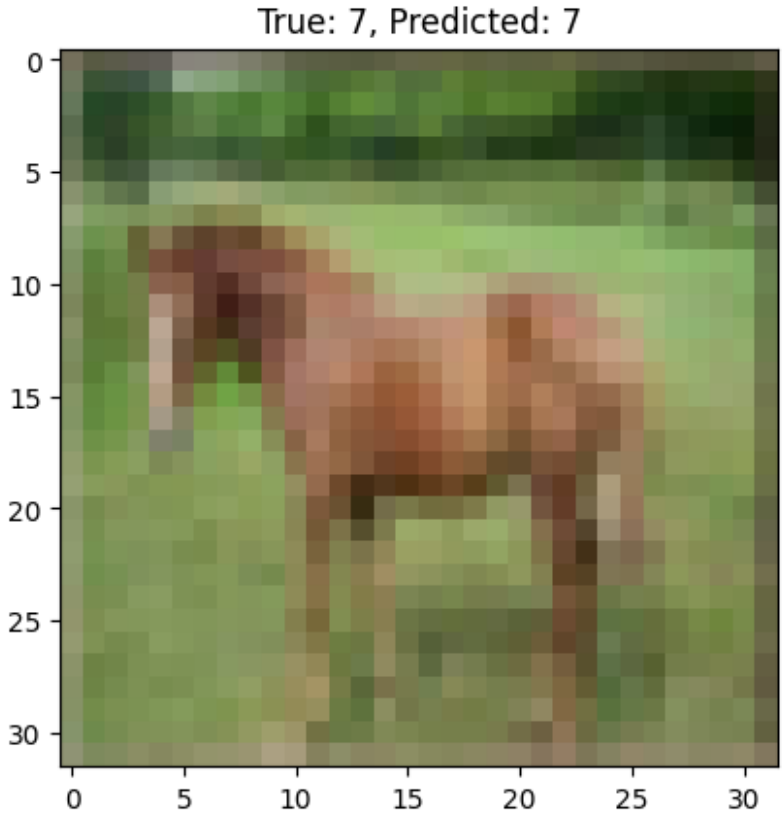

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

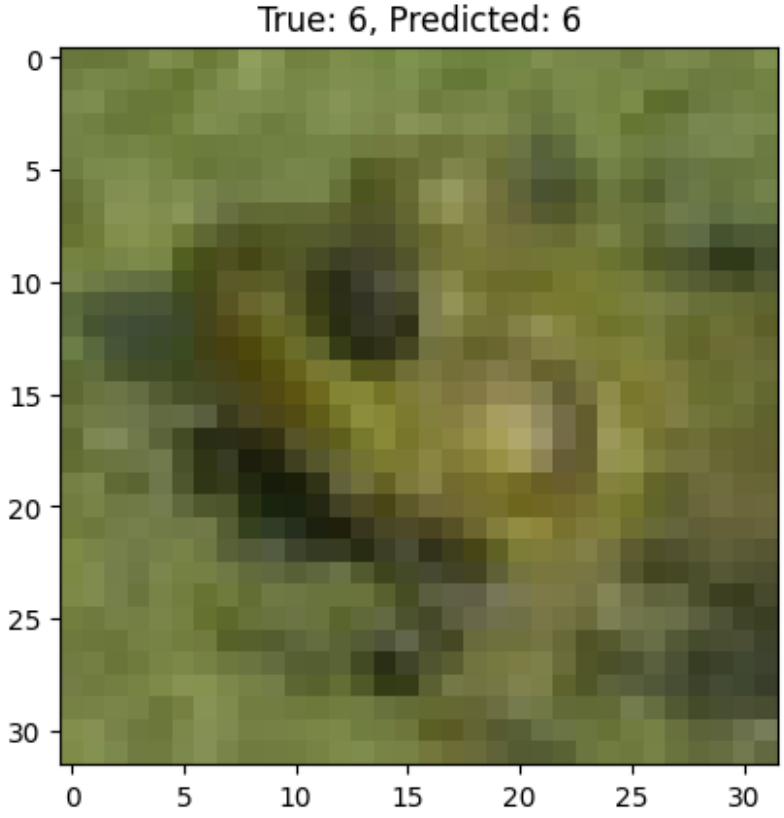

True: 0, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30

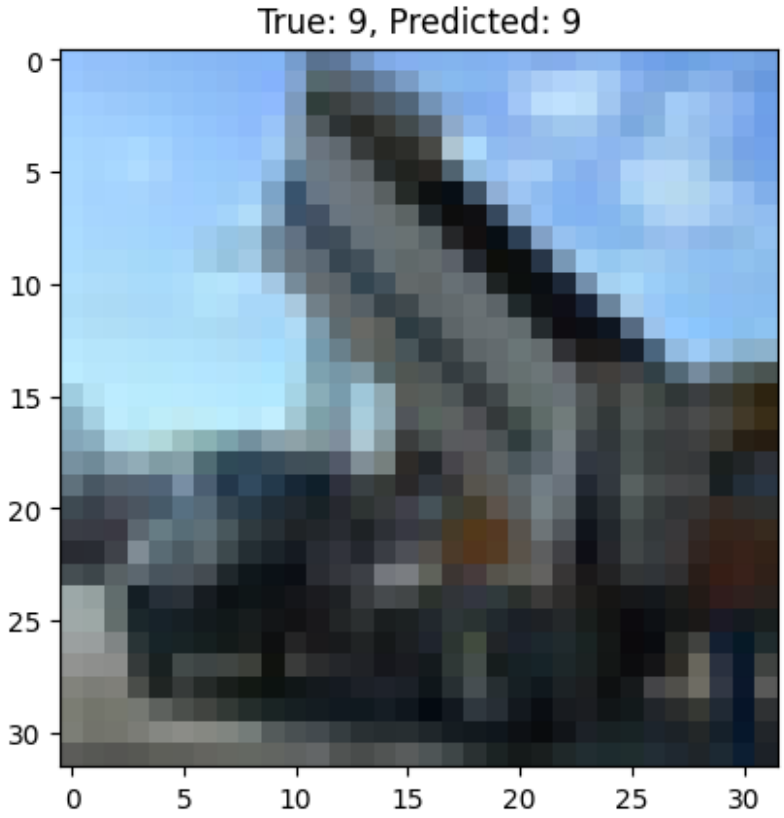

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

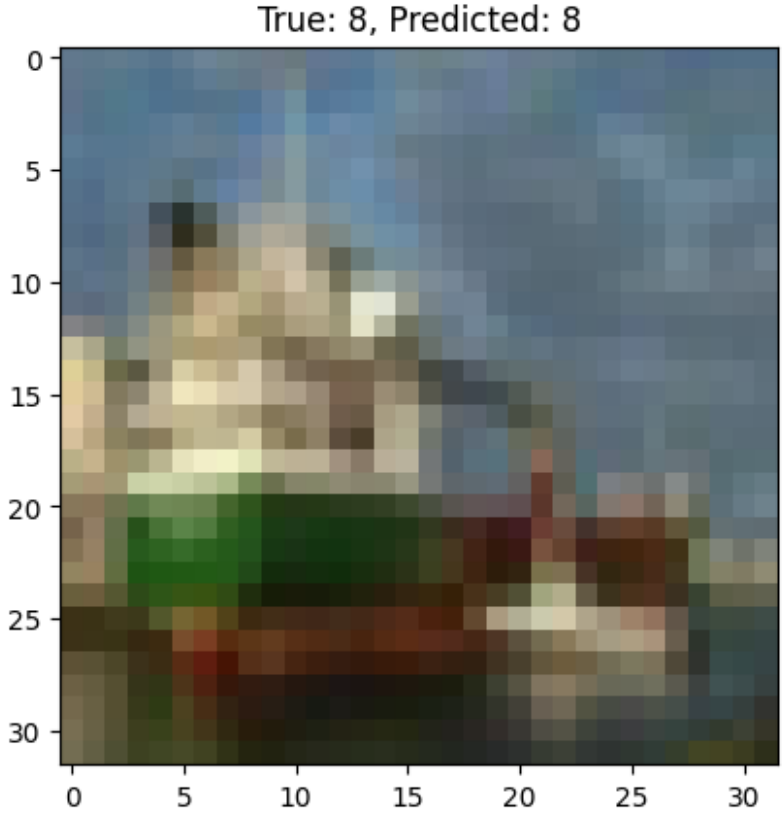

True: 3, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

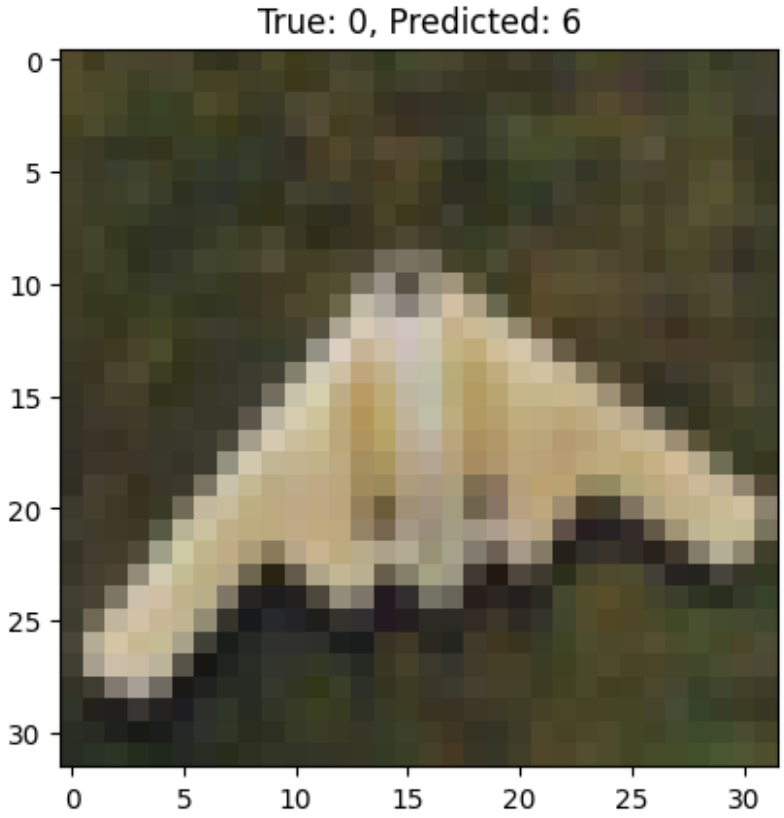

True: 9, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

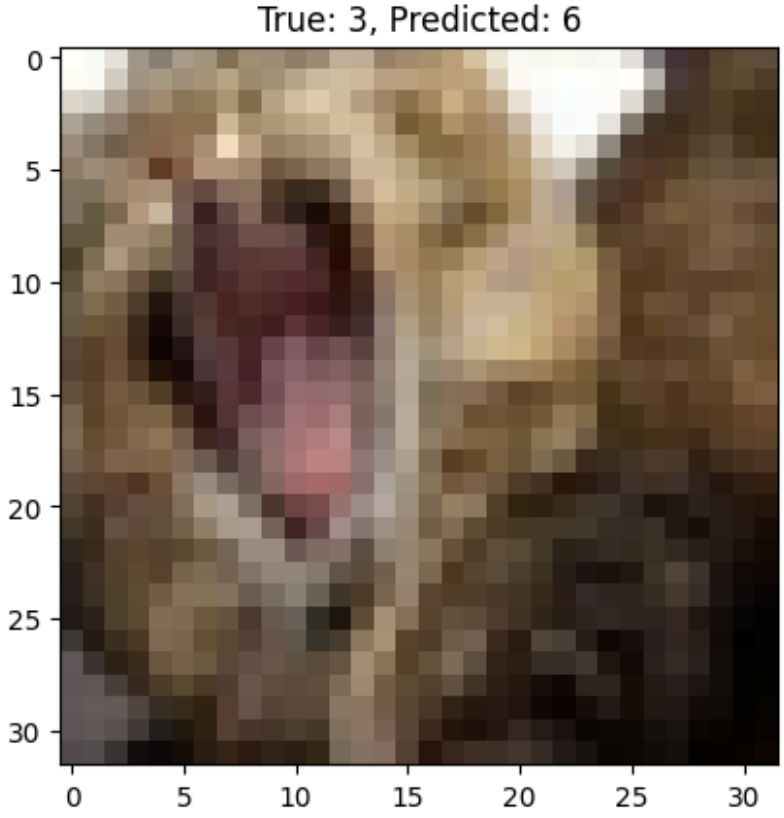

True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

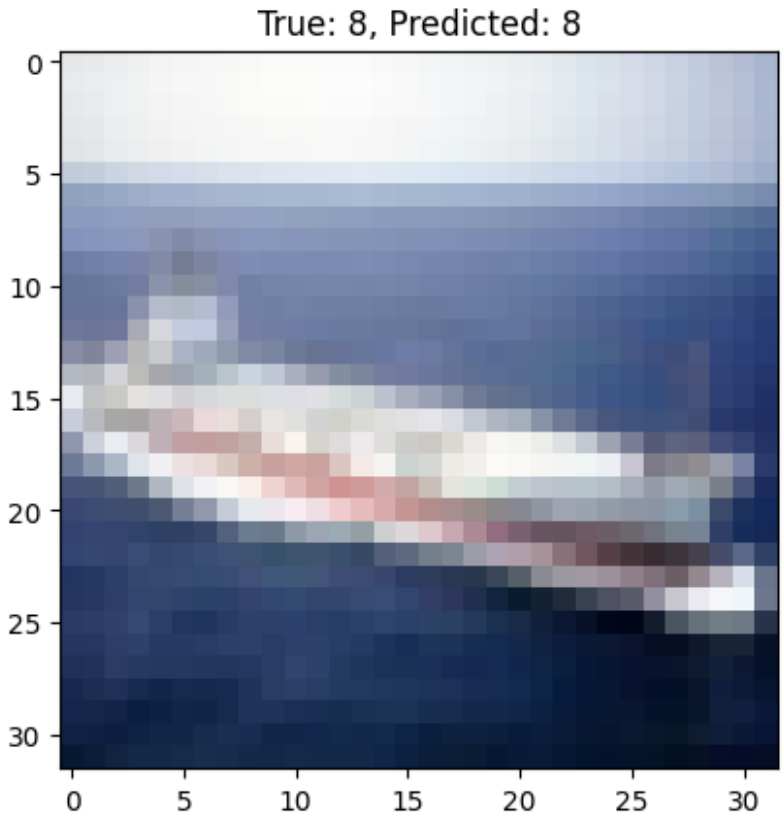

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

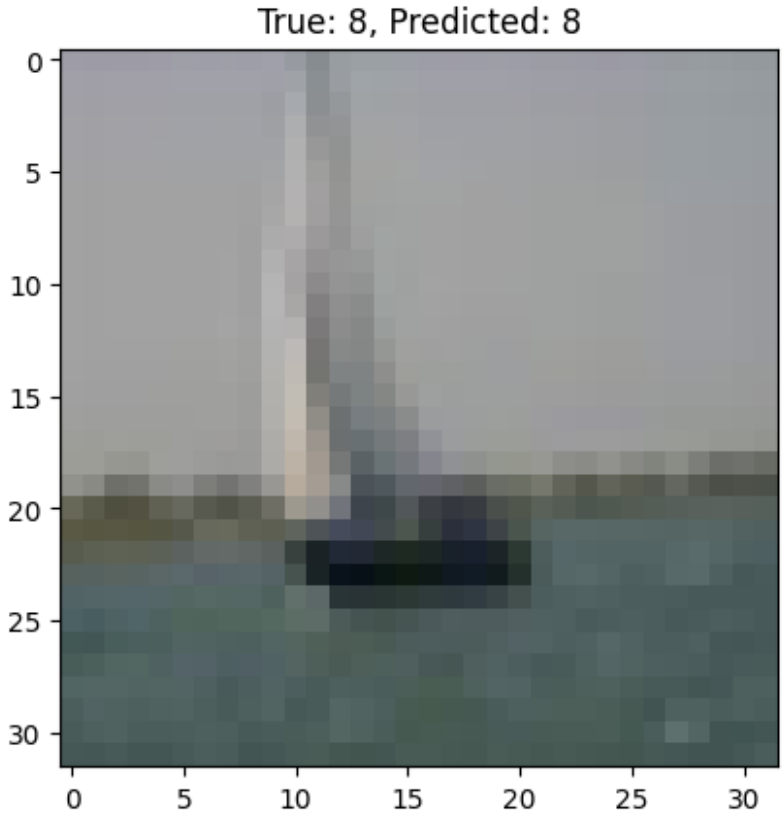

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

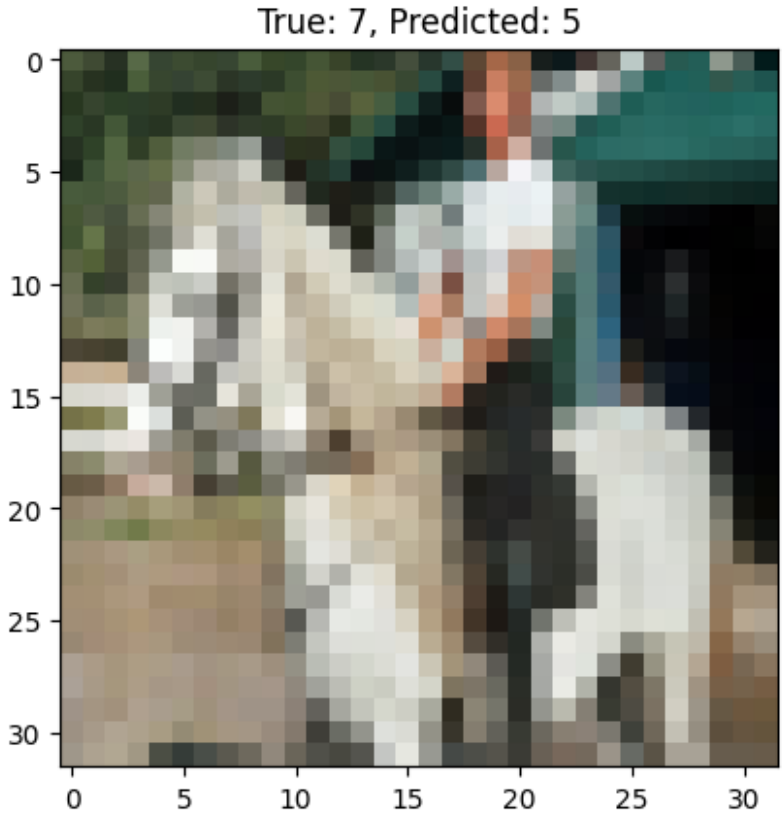

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

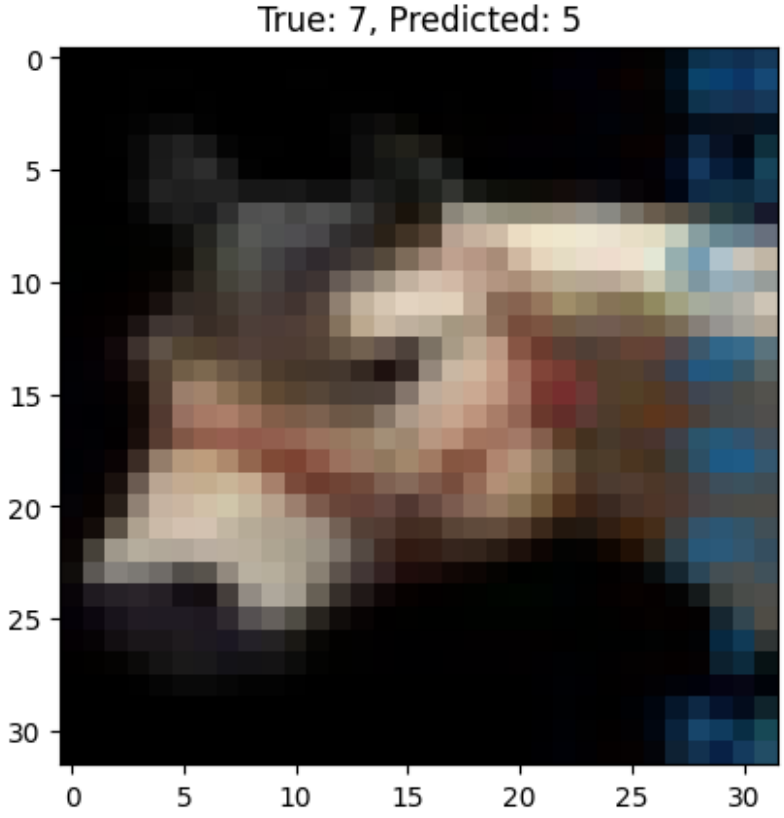

True: 0, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

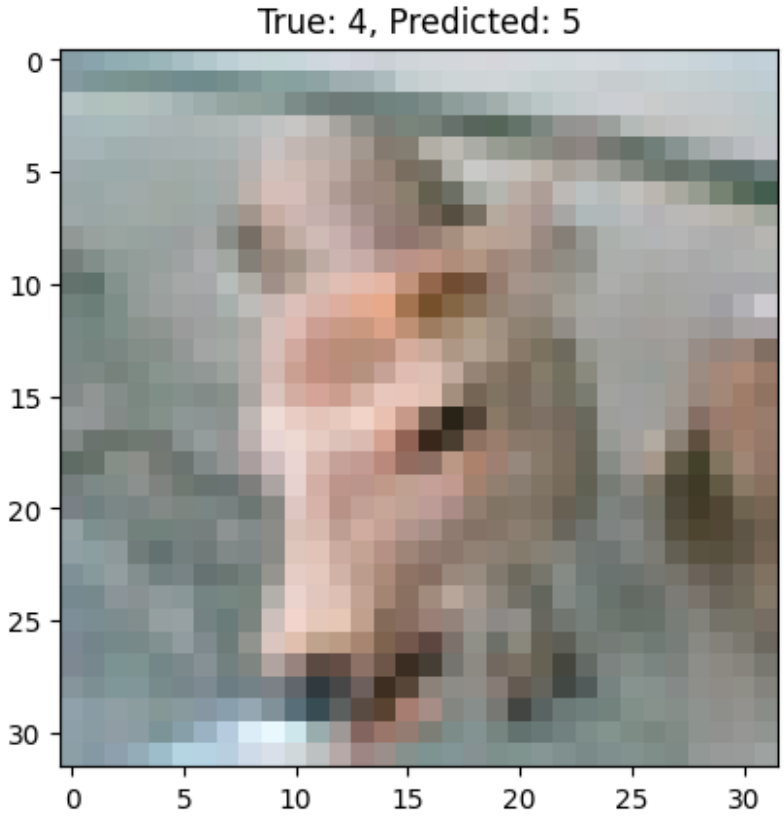

True: 3, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

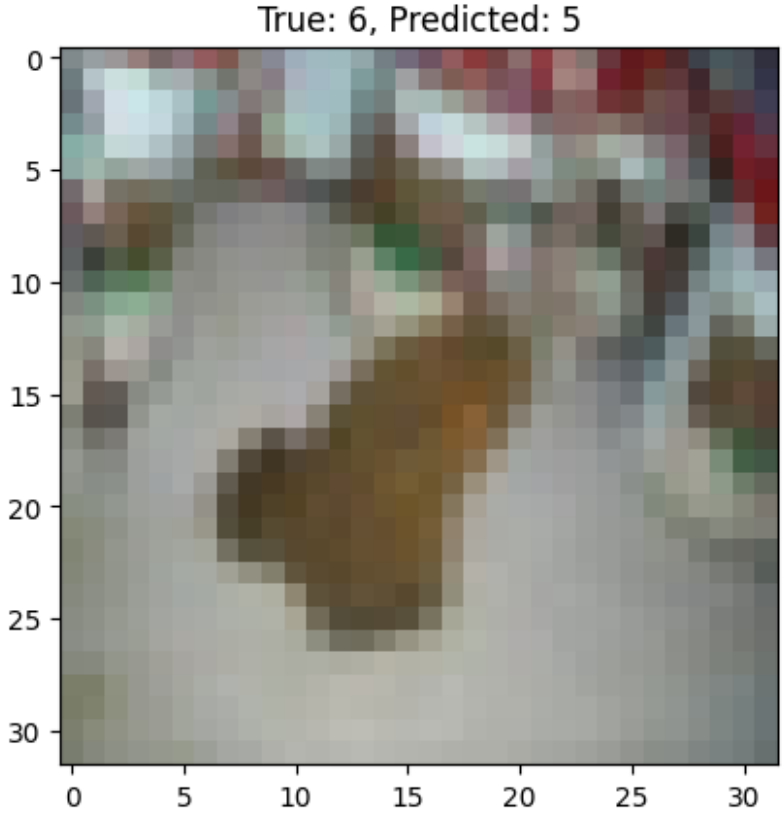

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

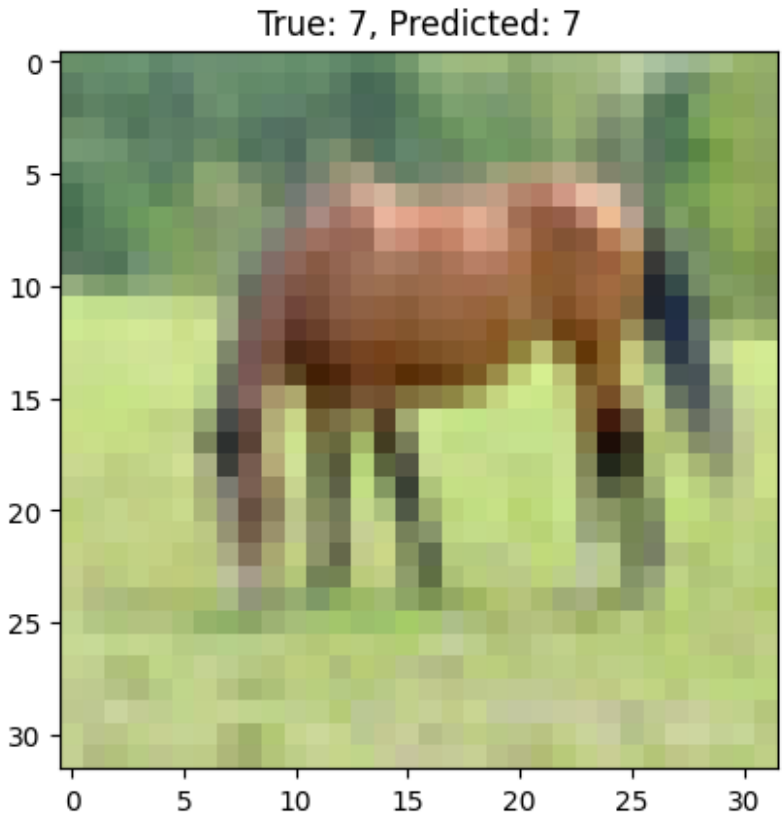

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

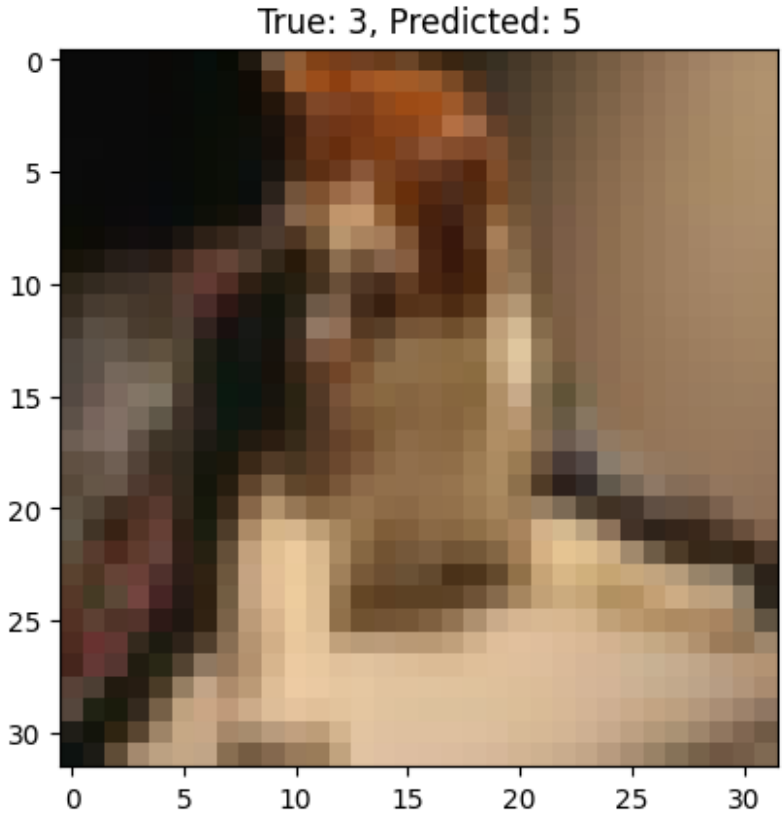

True: 7, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

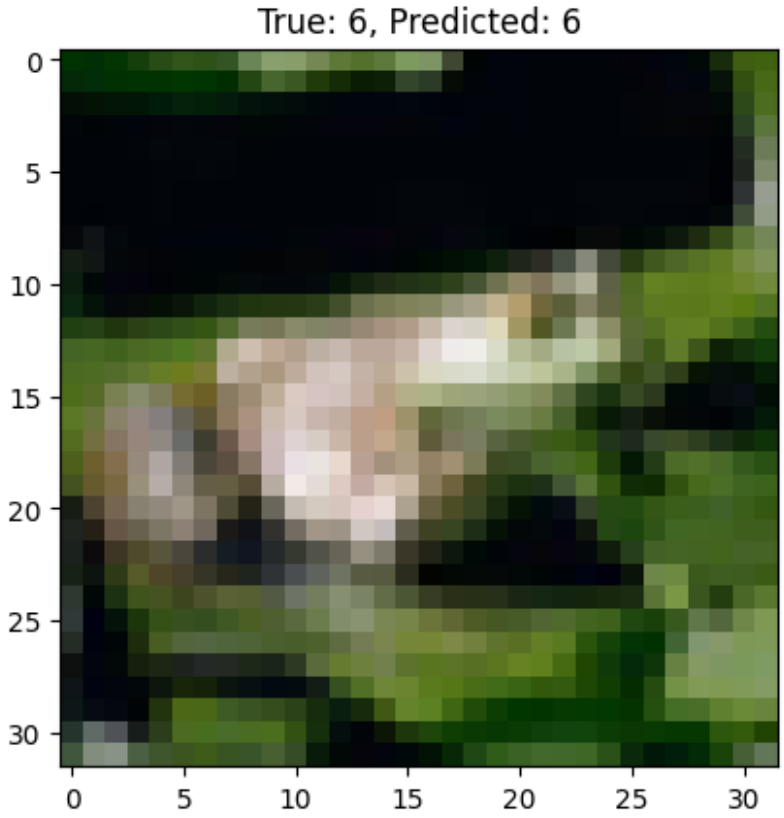

True: 7, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

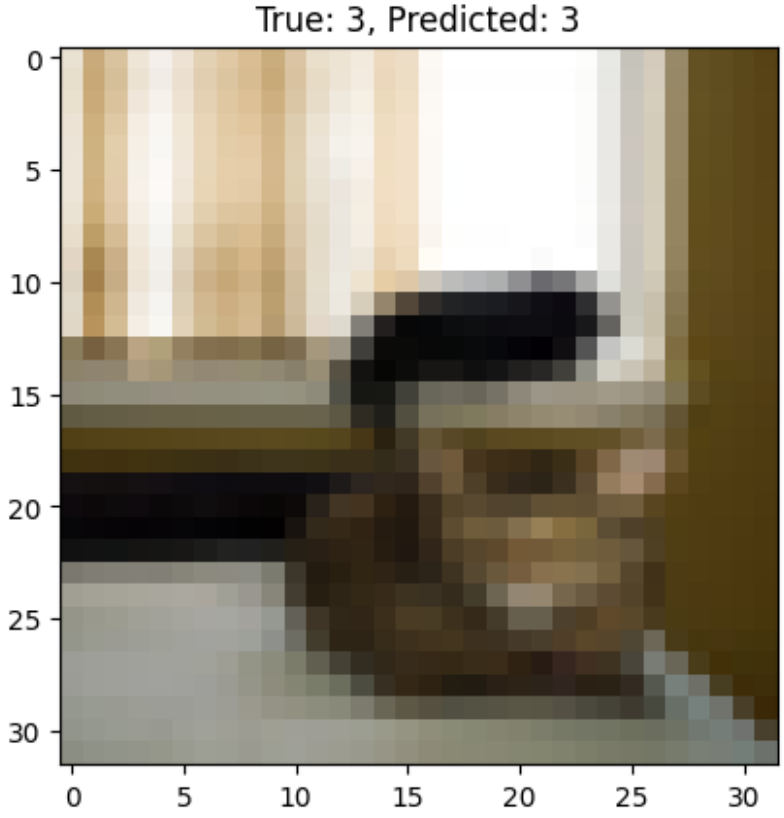

True: 4, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

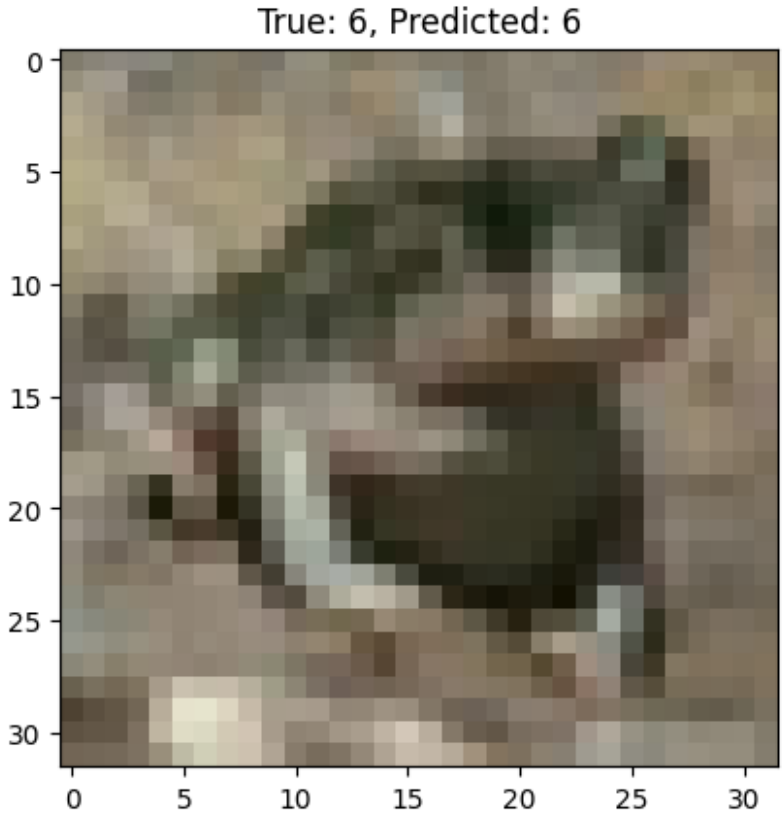

True: 6, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

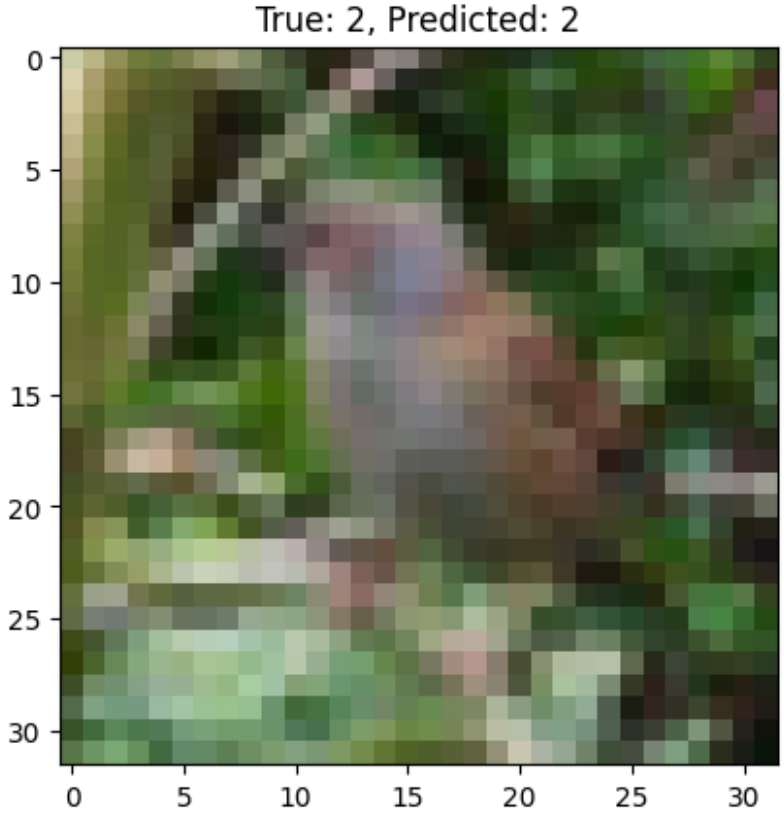

True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

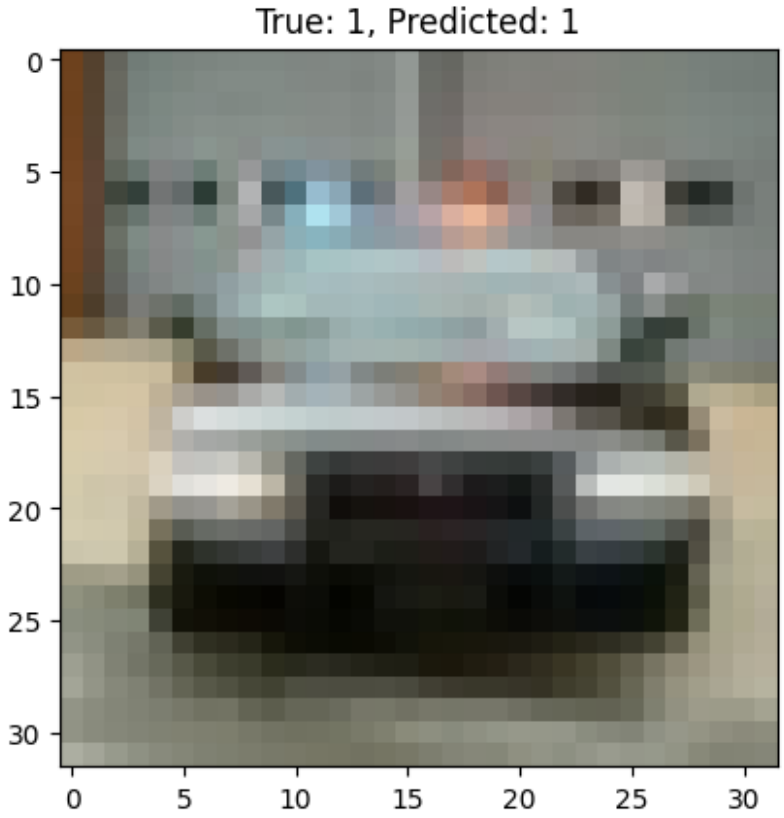

True: 3, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

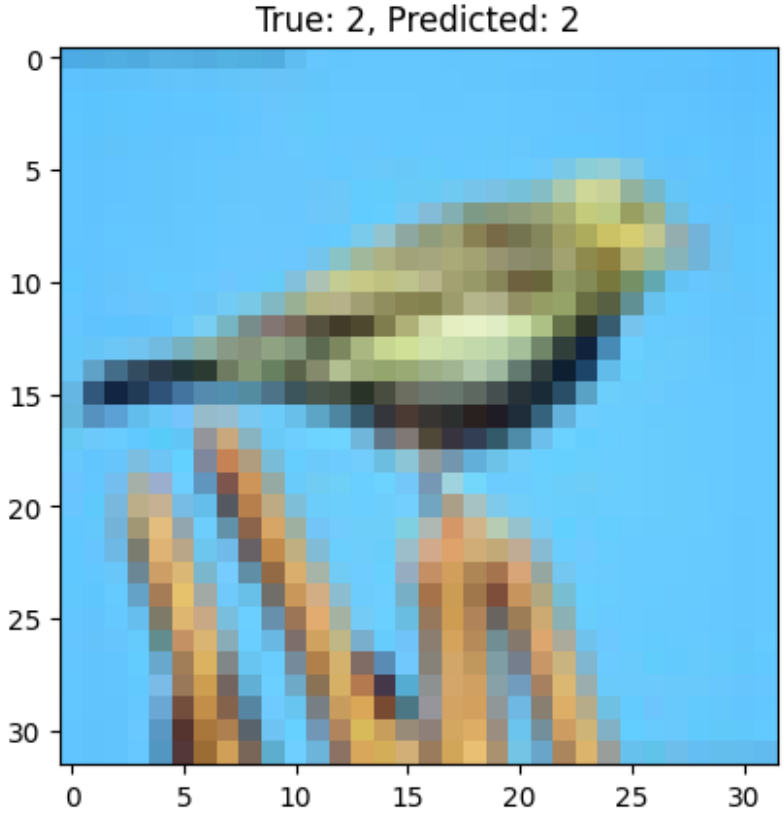

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

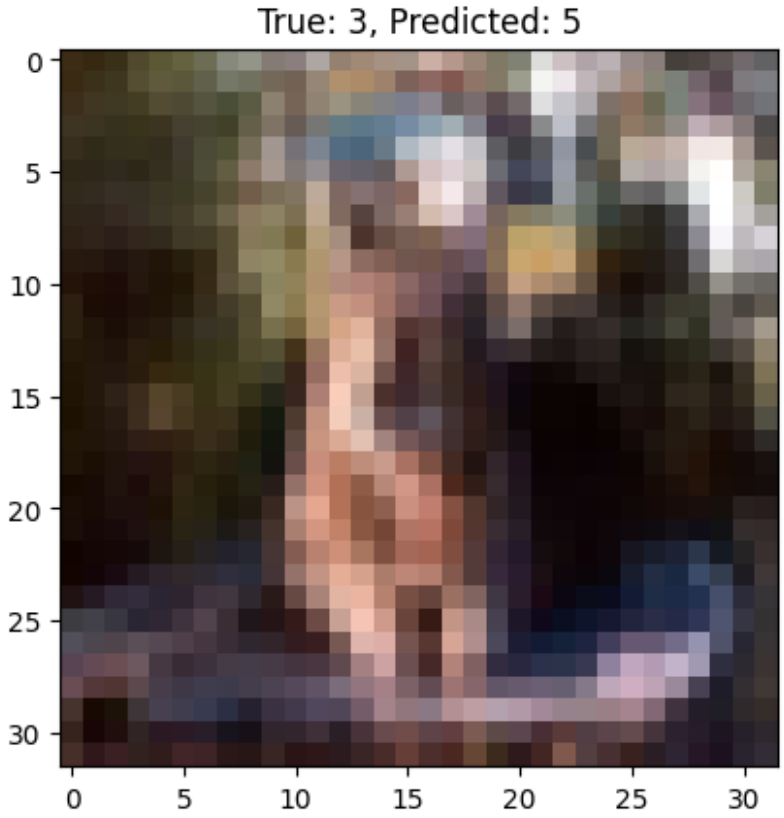

True: 3, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

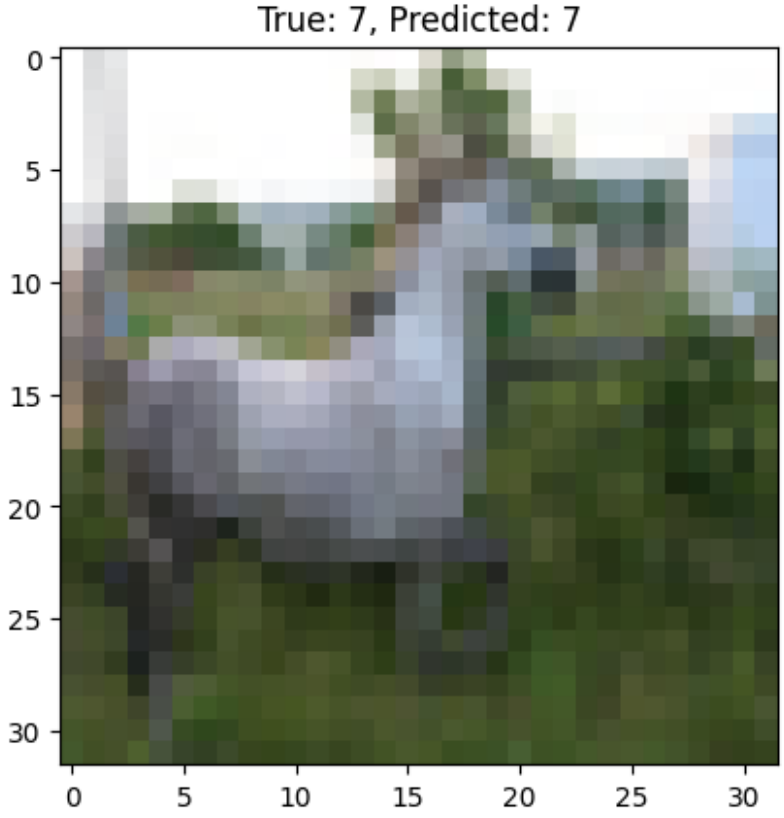

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

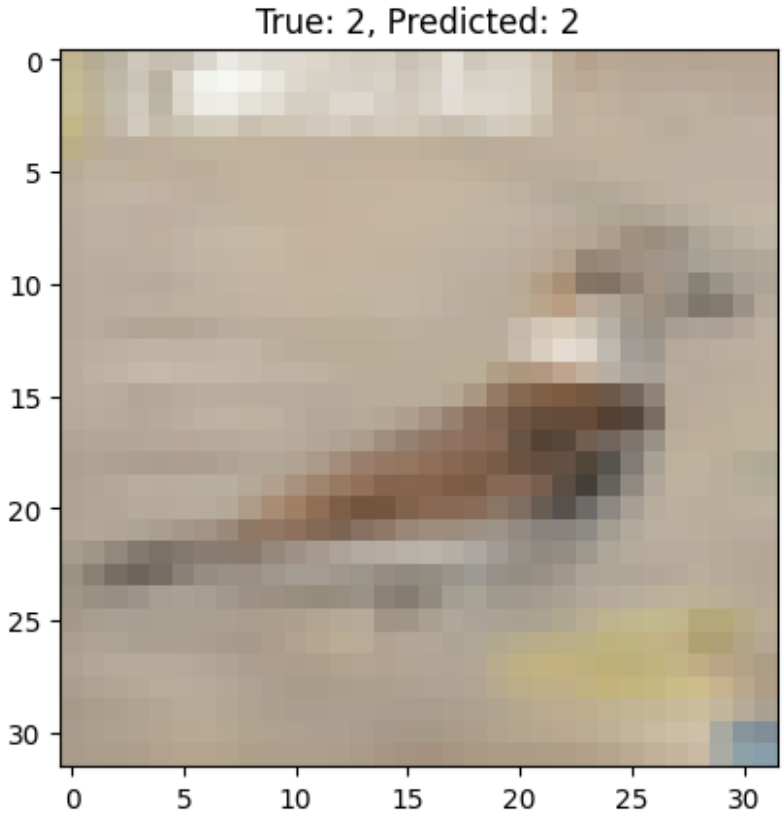

True: 2, Predicted: 2
0
5
10
15
20
25
30
0
5
10
15
20
25
30

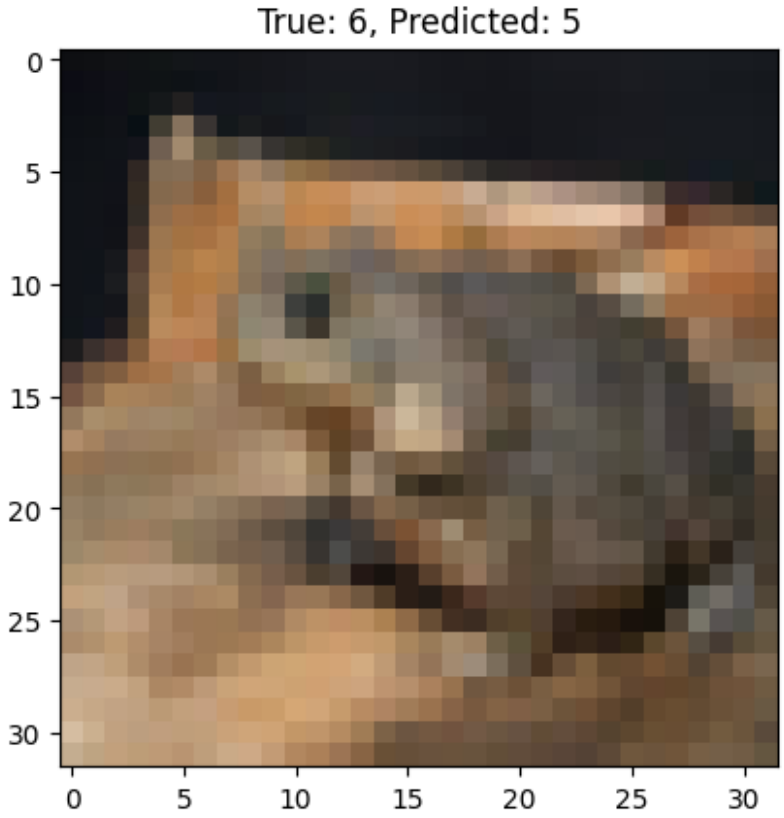

True: 1, Predicted: 1
0
5
10
15
20
25
30
0
5
10
15
20
25
30

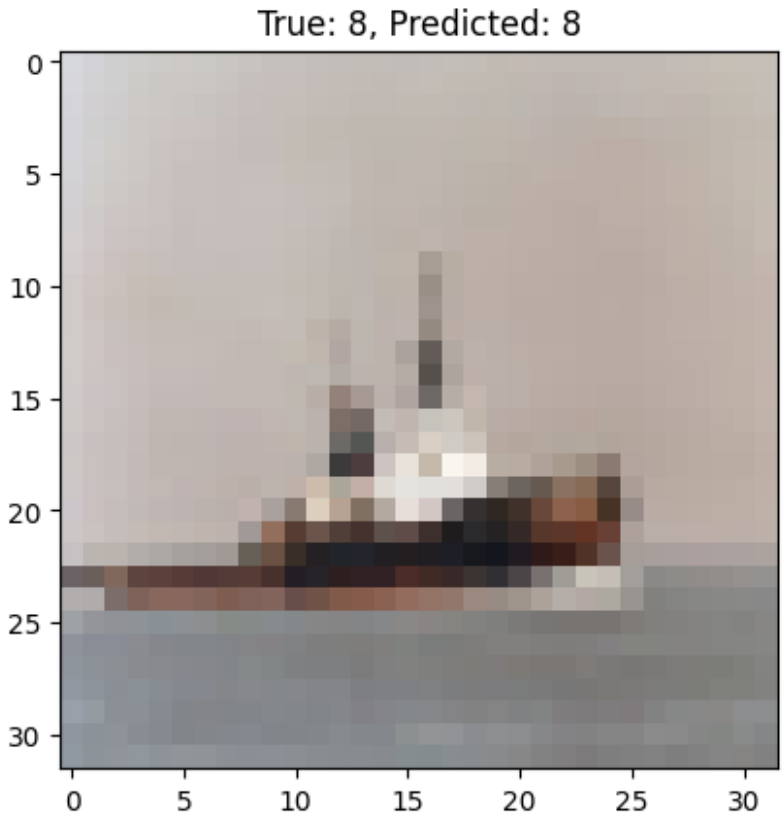

True: 2, Predicted: 2
0
5
10
15
20
25
30
0
5
10
15
20
25
30

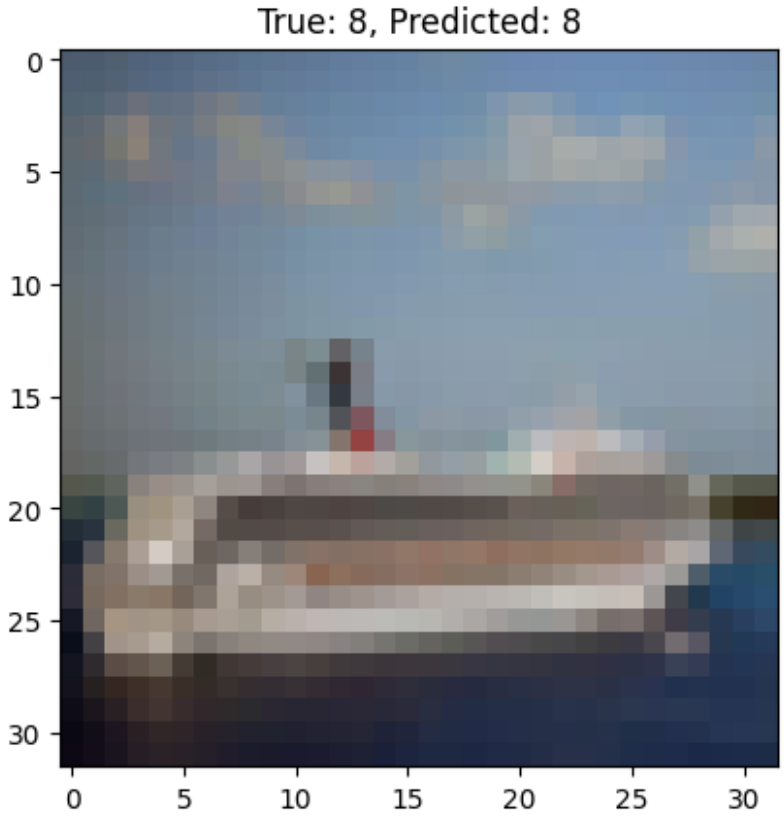

True: 3, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

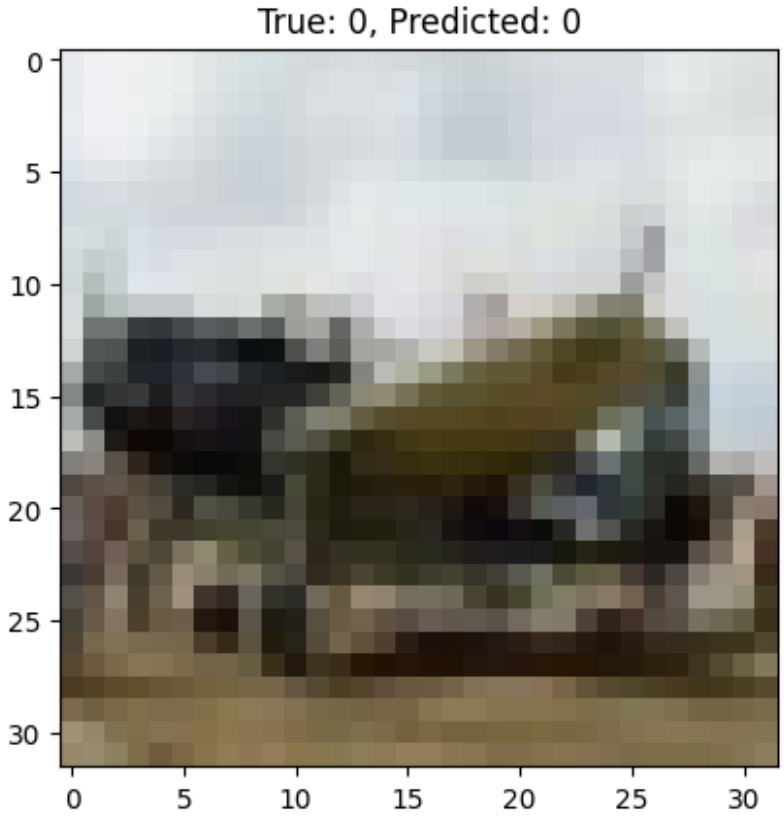

True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

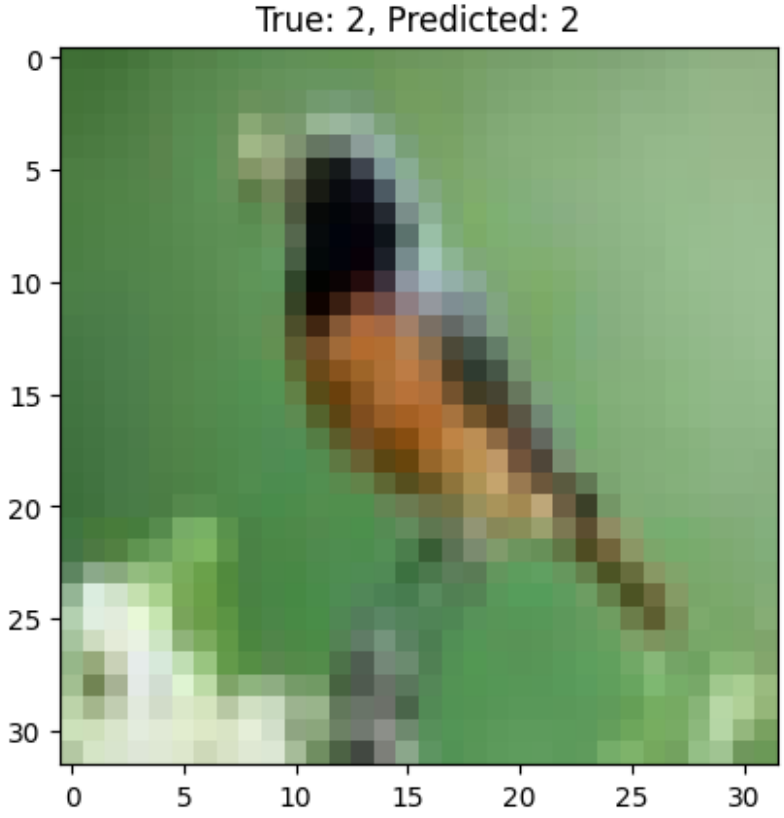

True: 2, Predicted: 2
0
5
10
15
20
25
30
0
5
10
15
20
25
30

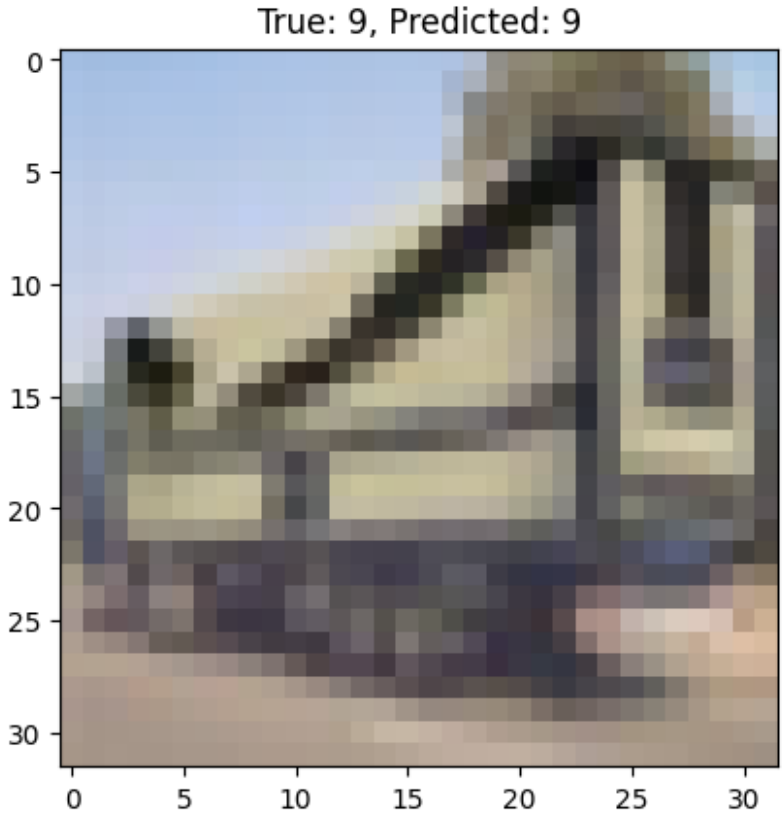

True: 6, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

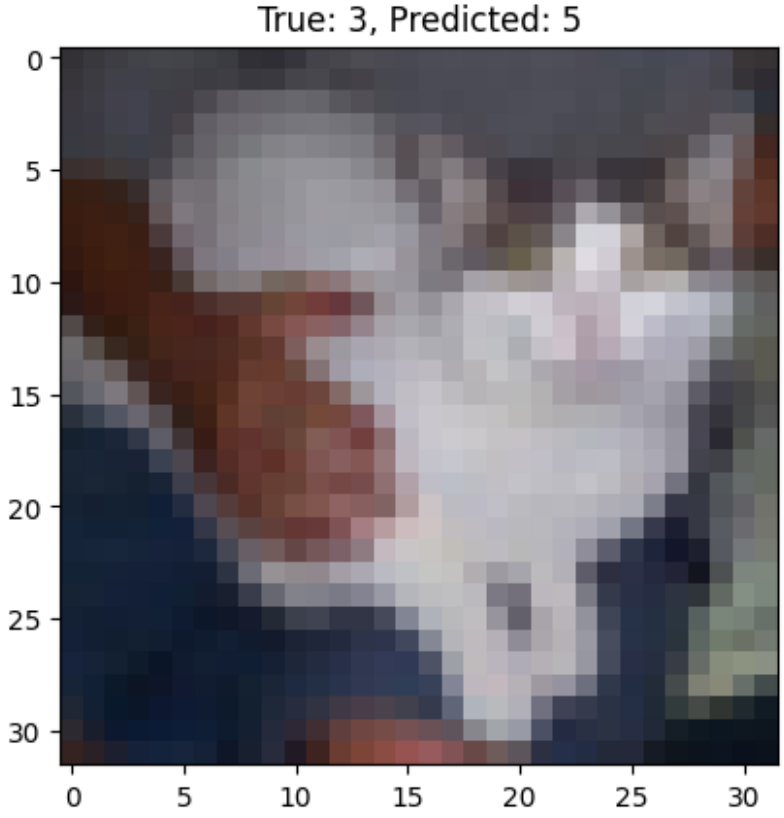

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

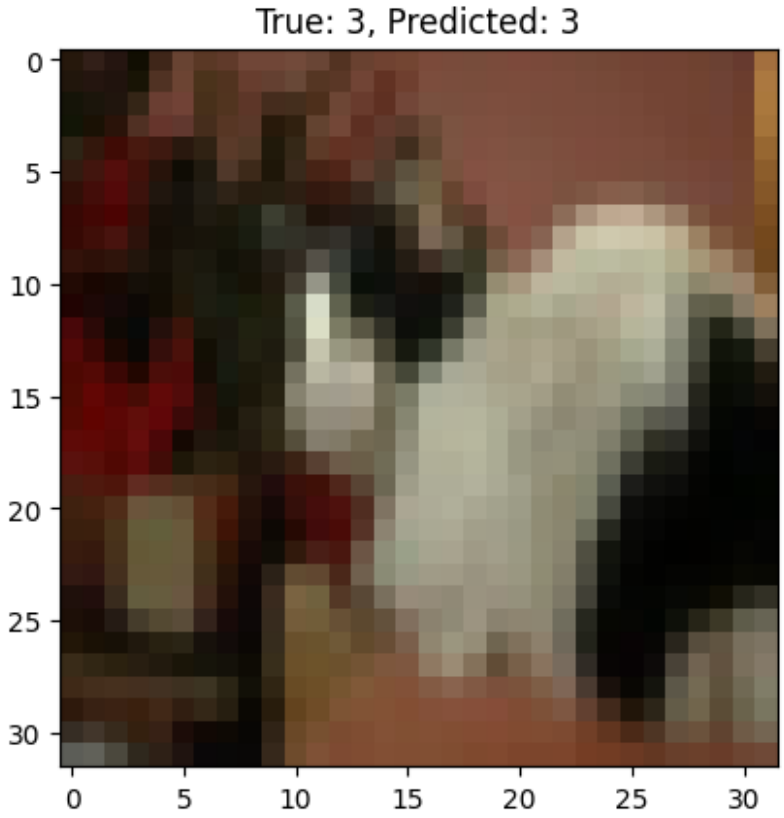

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

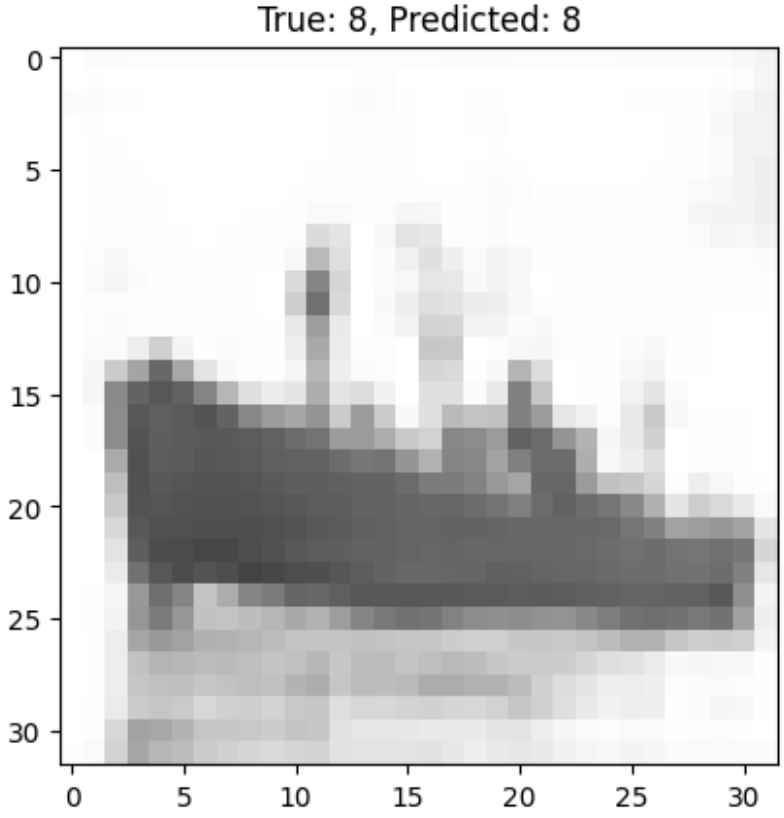

True: 0, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30

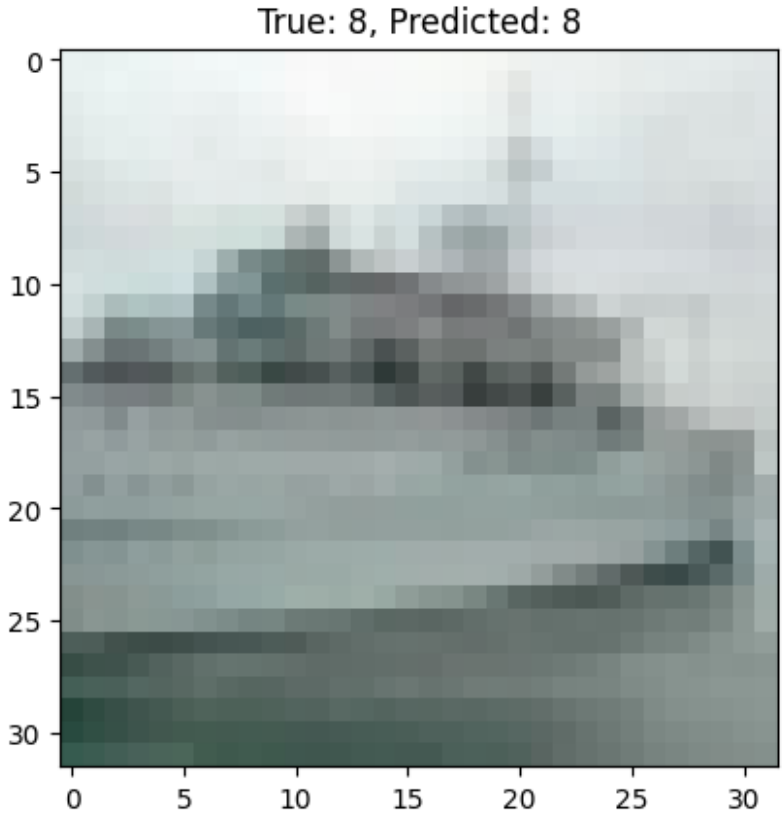

True: 2, Predicted: 2
0
5
10
15
20
25
30
0
5
10
15
20
25
30

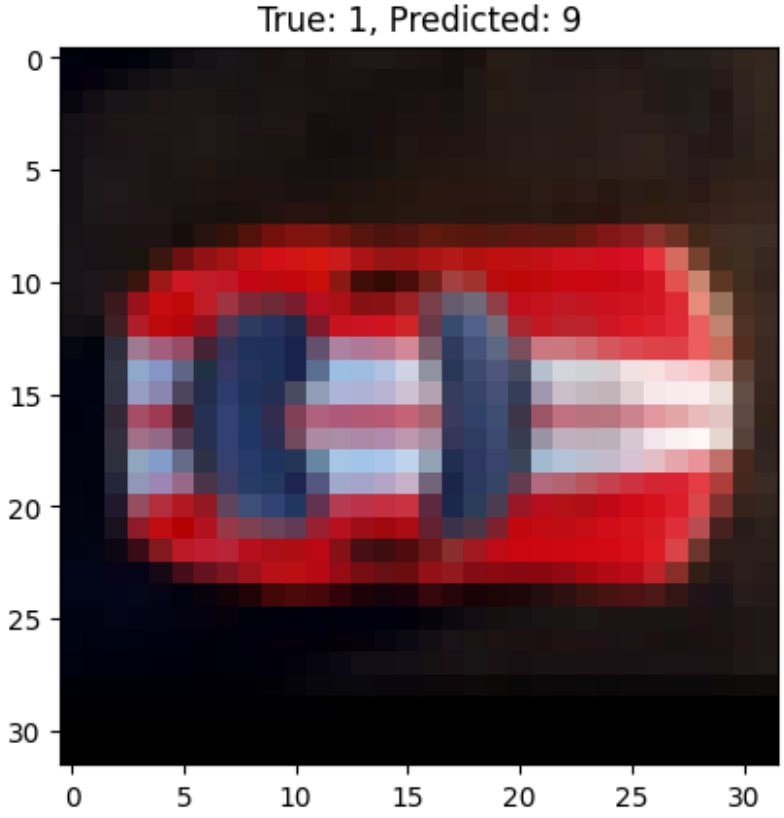

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

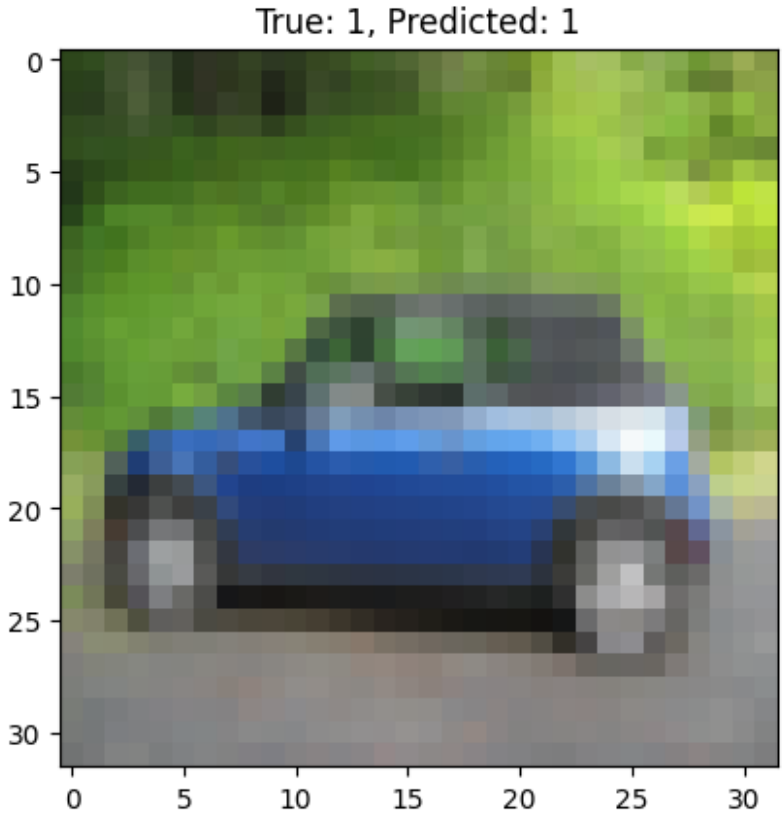

True: 3, Predicted: 5
0
5
10
15
20
25
30
0
5
10
15
20
25
30

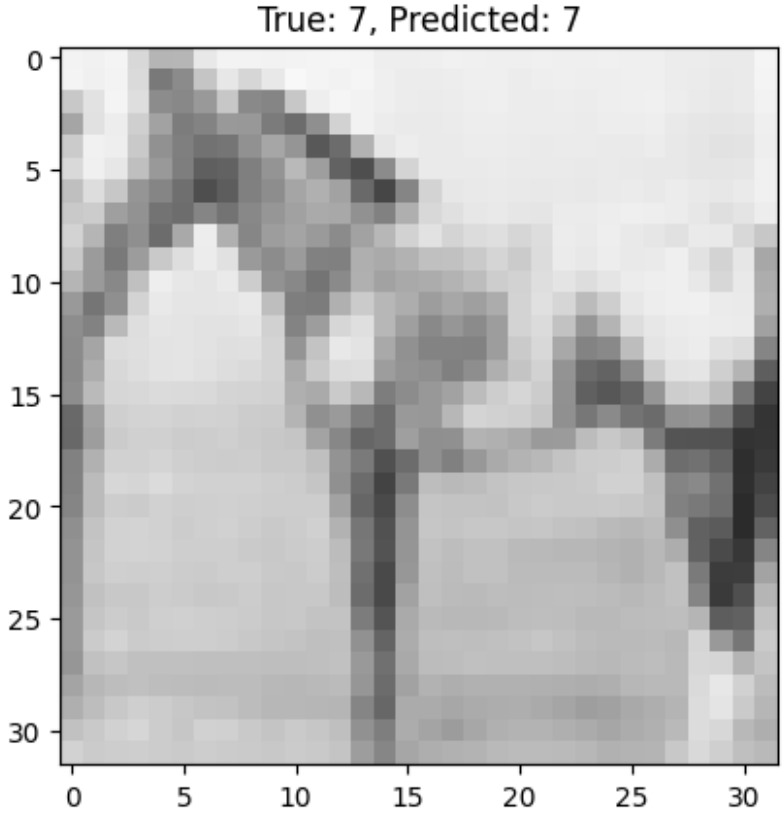

True: 3, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

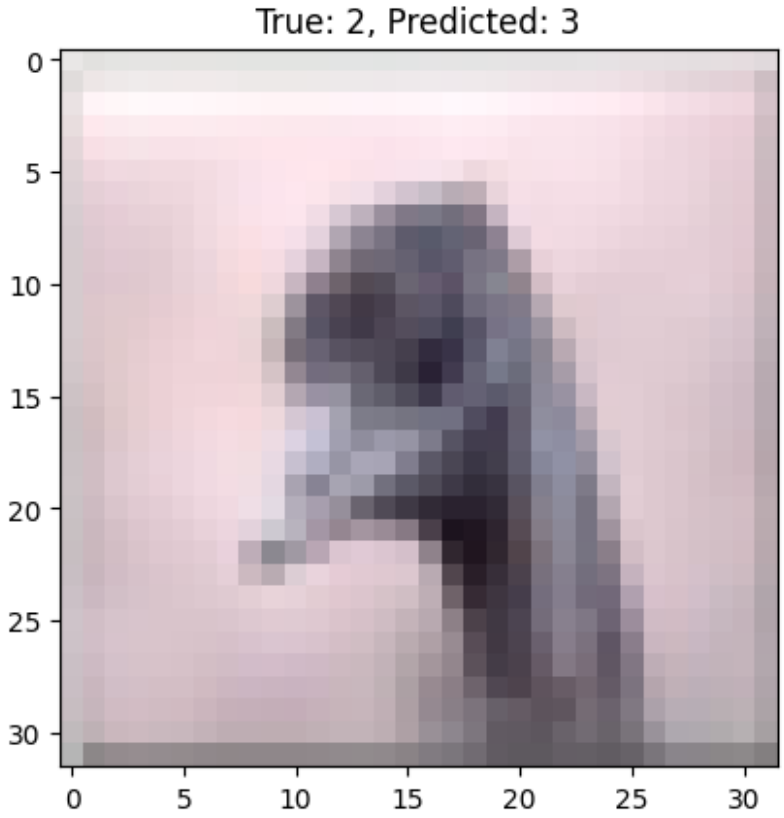

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

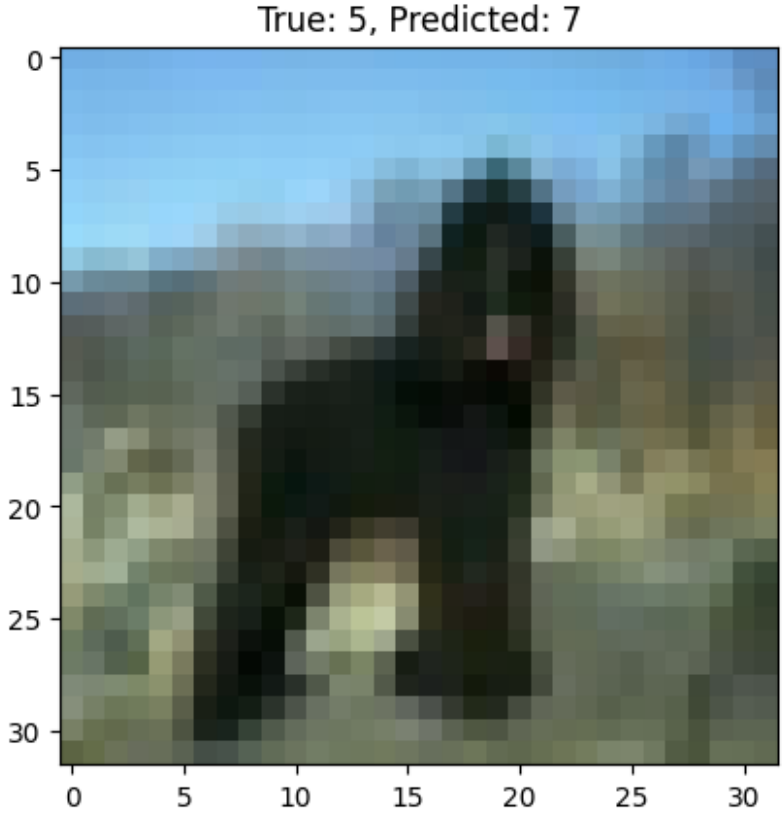

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

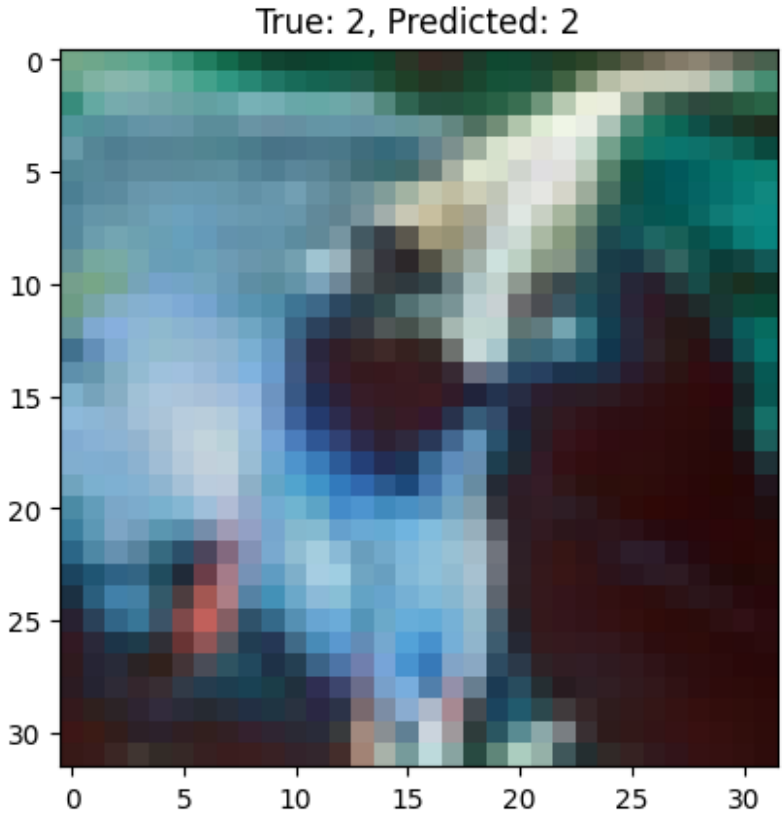

True: 1, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

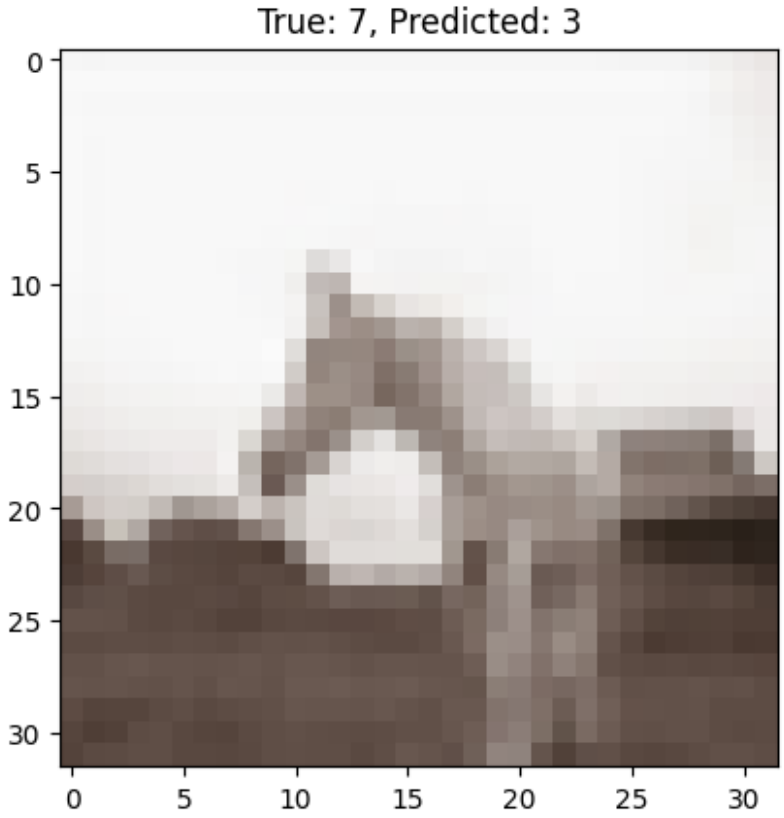

True: 1, Predicted: 1
0
5
10
15
20
25
30
0
5
10
15
20
25
30

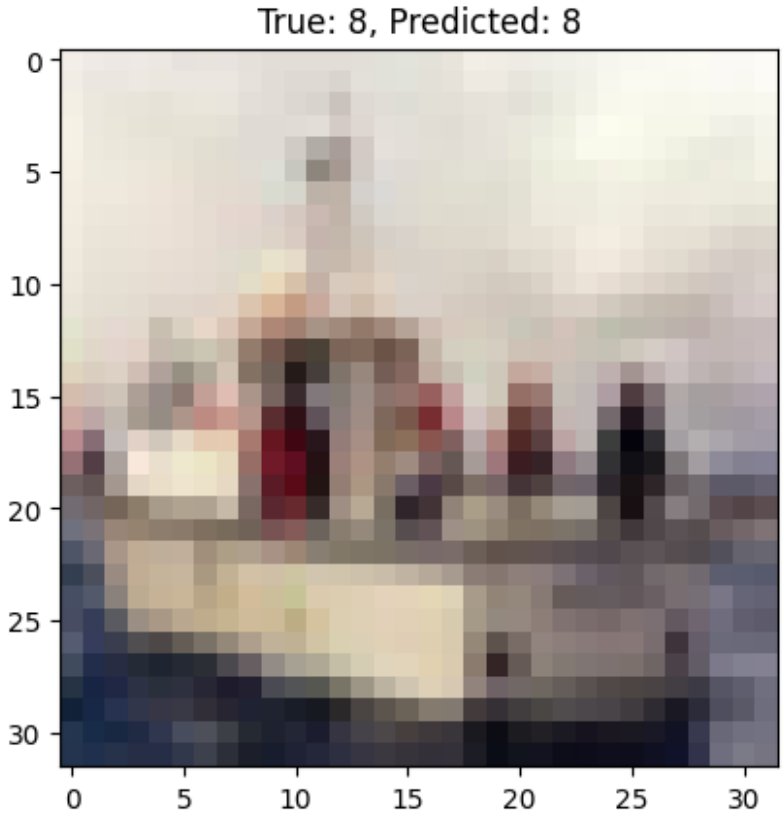

True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

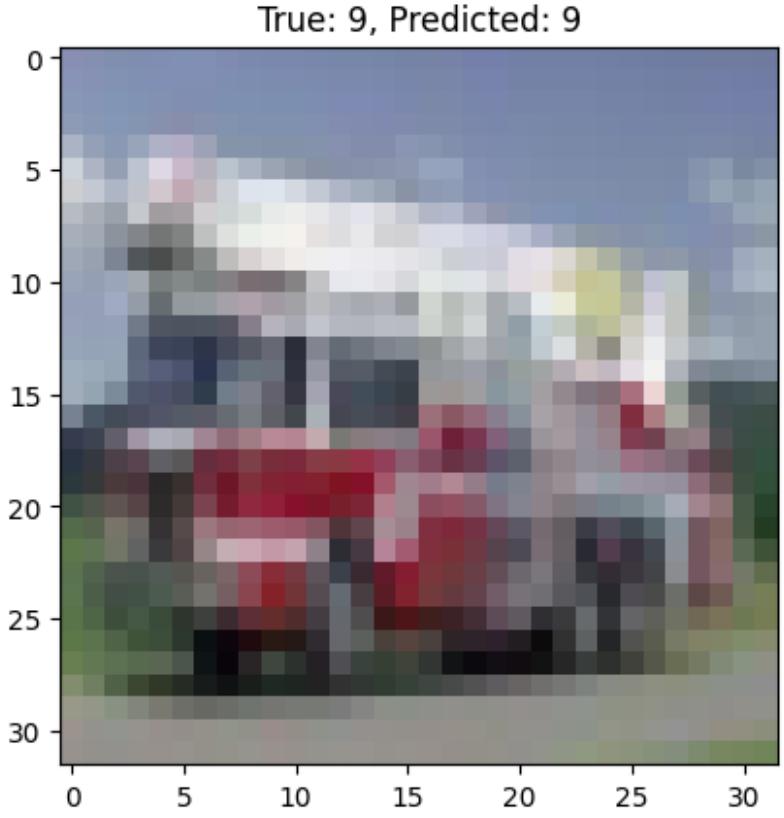

True: 2, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

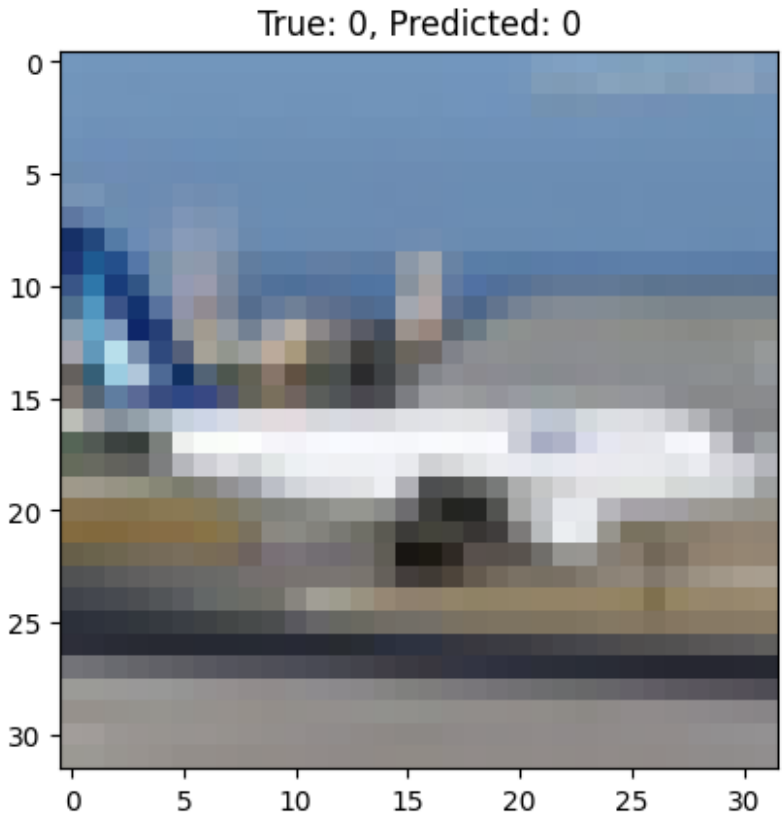

True: 5, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

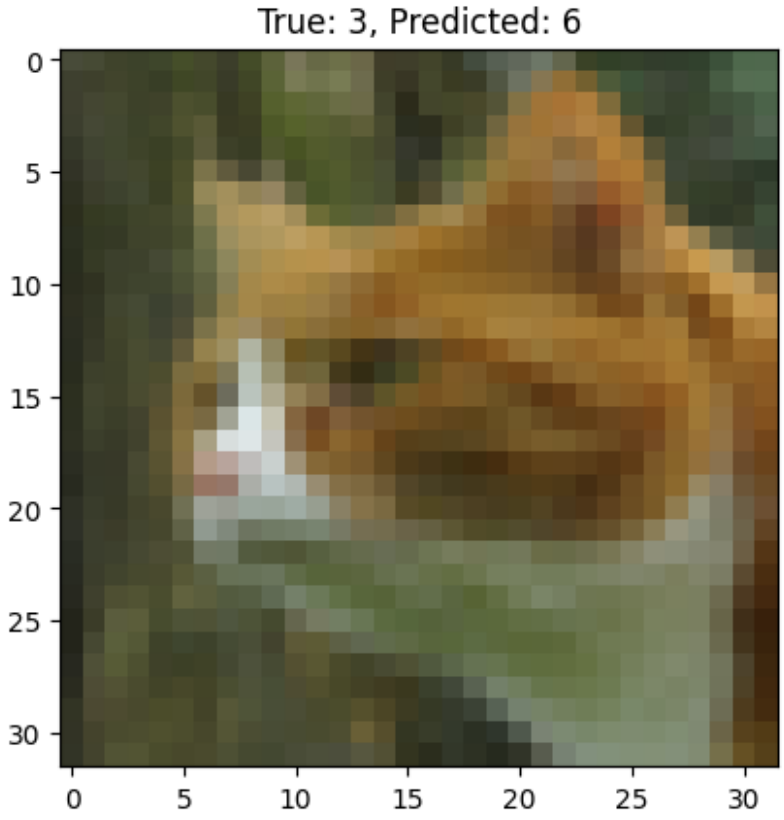

True: 2, Predicted: 2
0
5
10
15
20
25
30
0
5
10
15
20
25
30

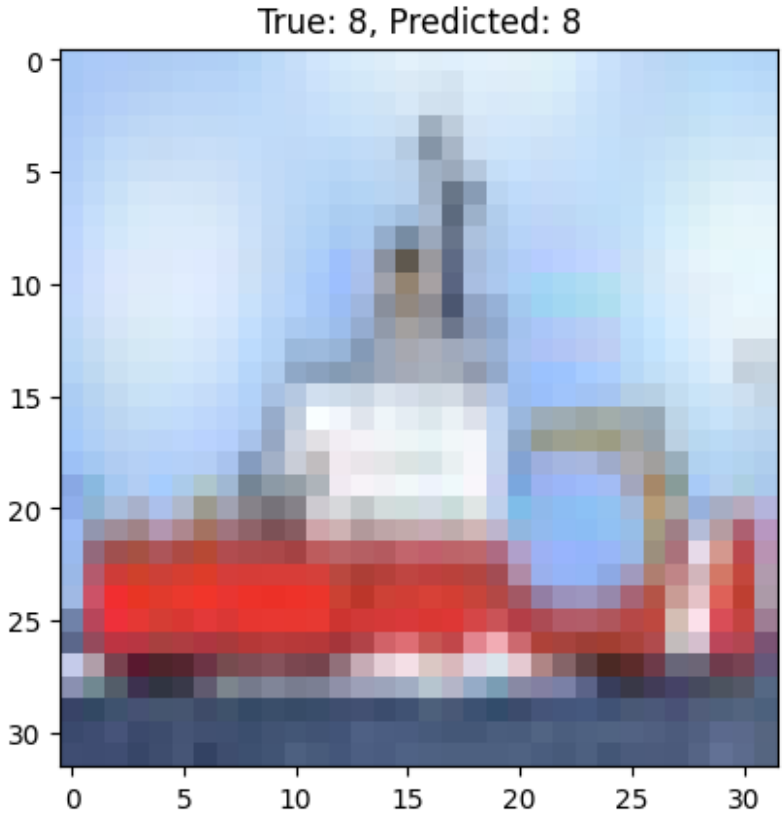

True: 7, Predicted: 3
0
5
10
15
20
25
30
0
5
10
15
20
25
30

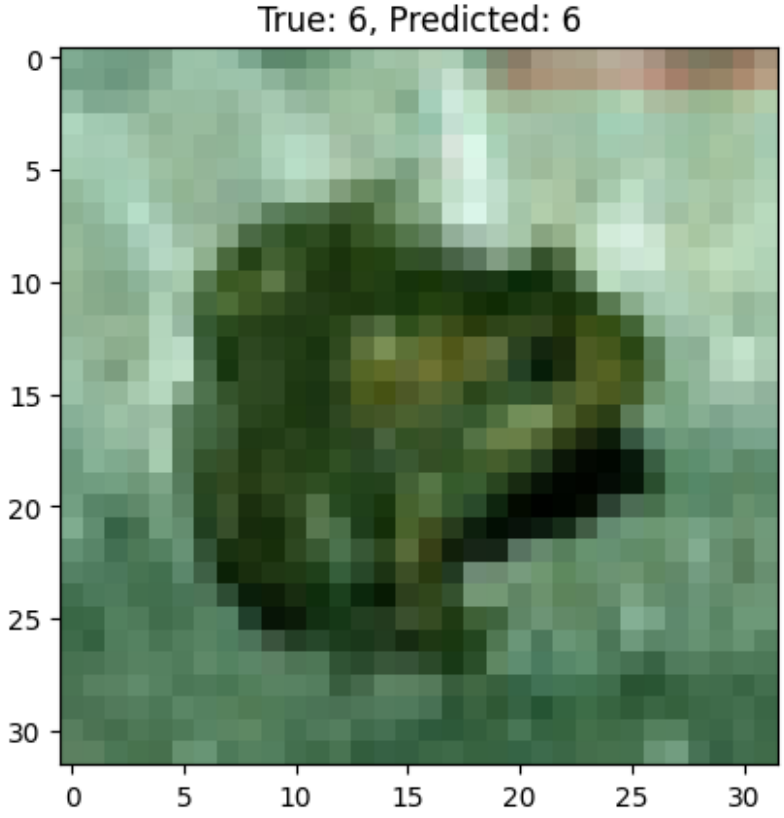

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

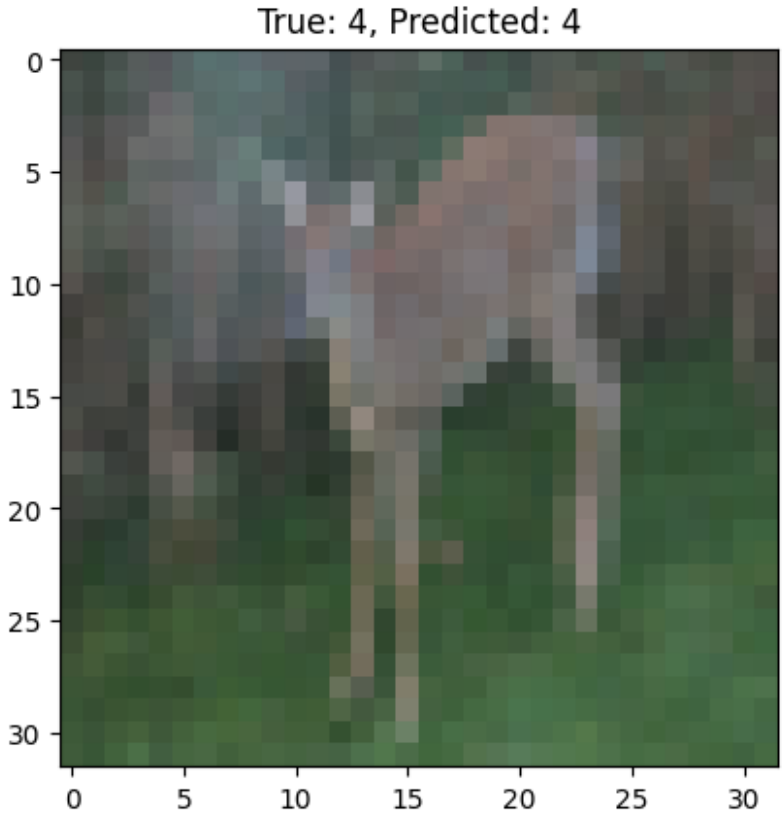

True: 9, Predicted: 9
0
5
10
15
20
25
30
0
5
10
15
20
25
30

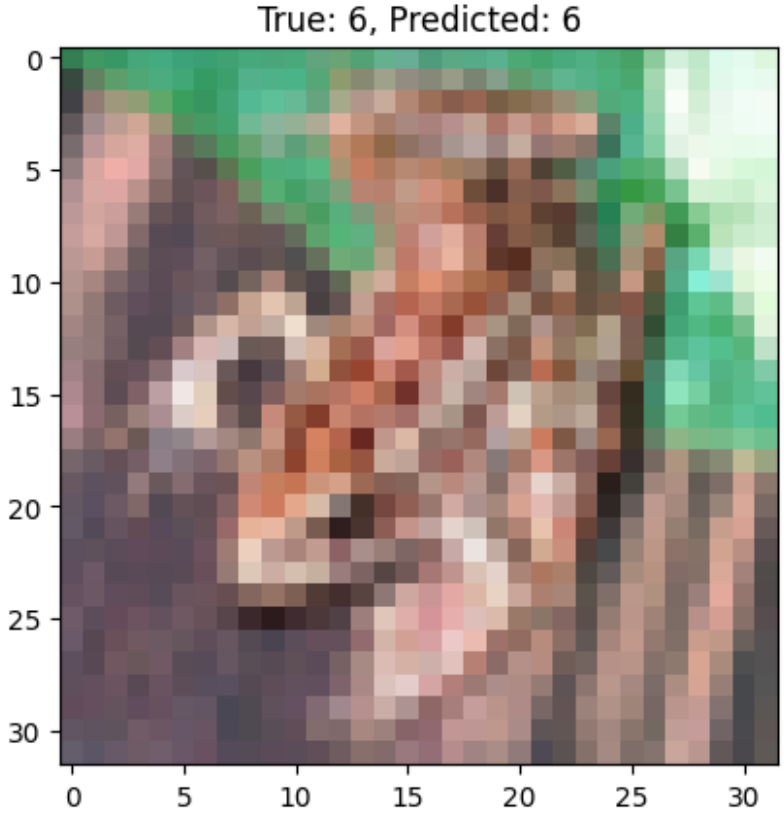

True: 0, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30

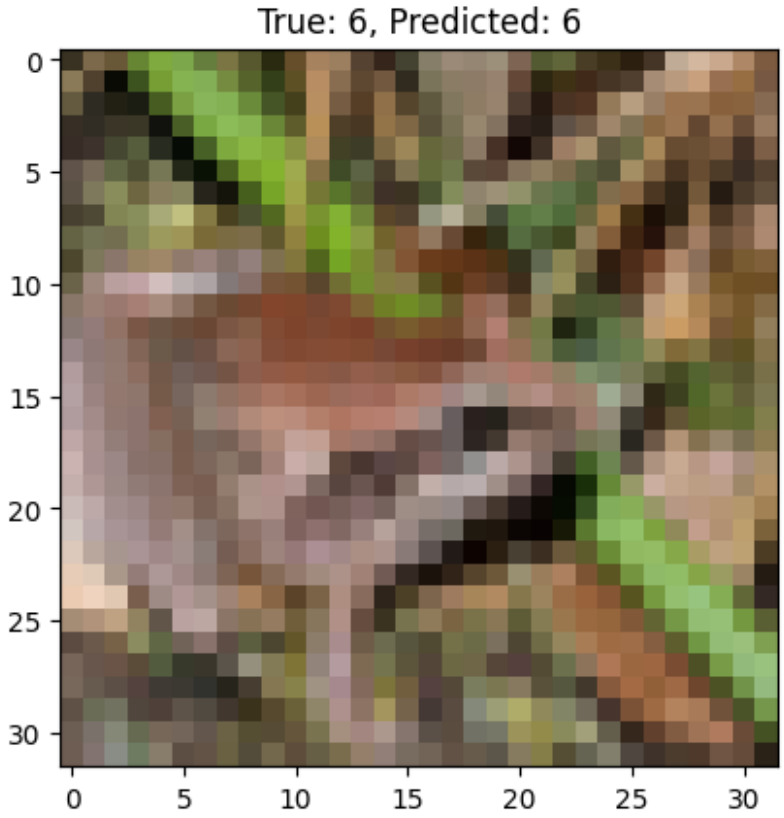

True: 3, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

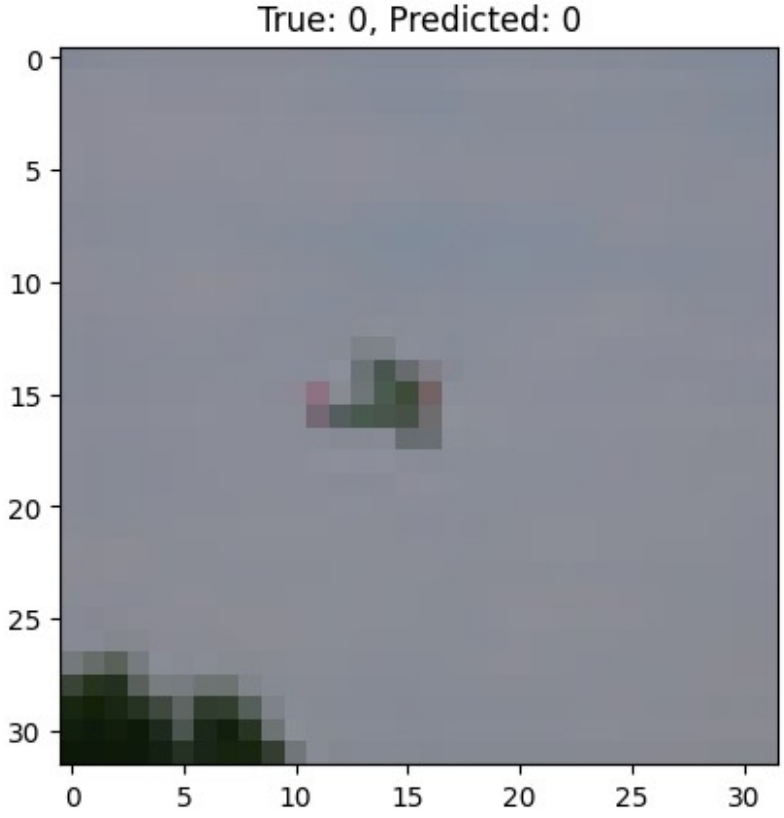

True: 8, Predicted: 8
0
5
10
15
20
25
30
0
5
10
15
20
25
30

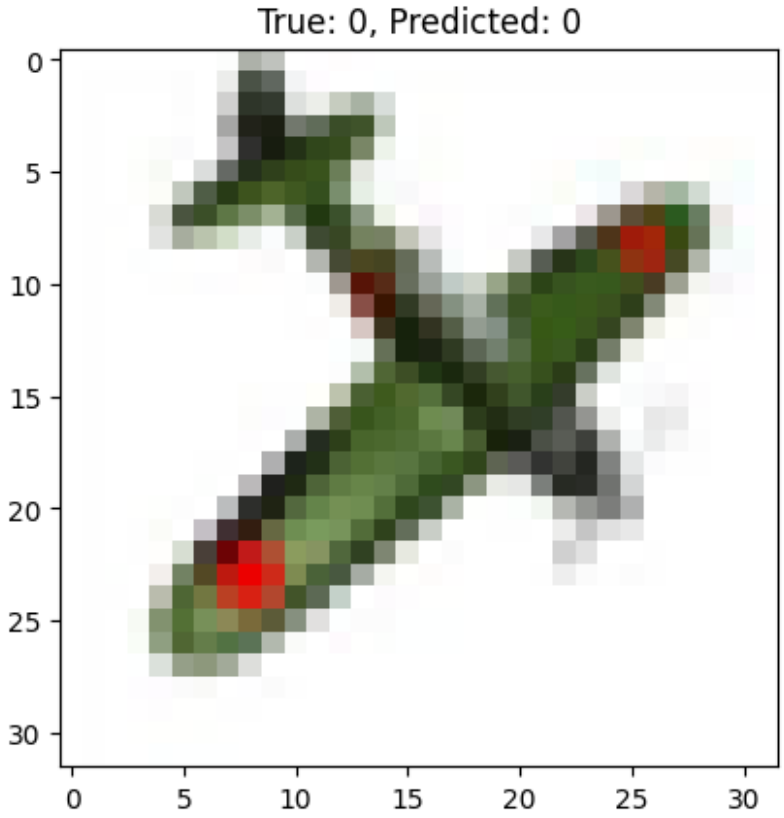

True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

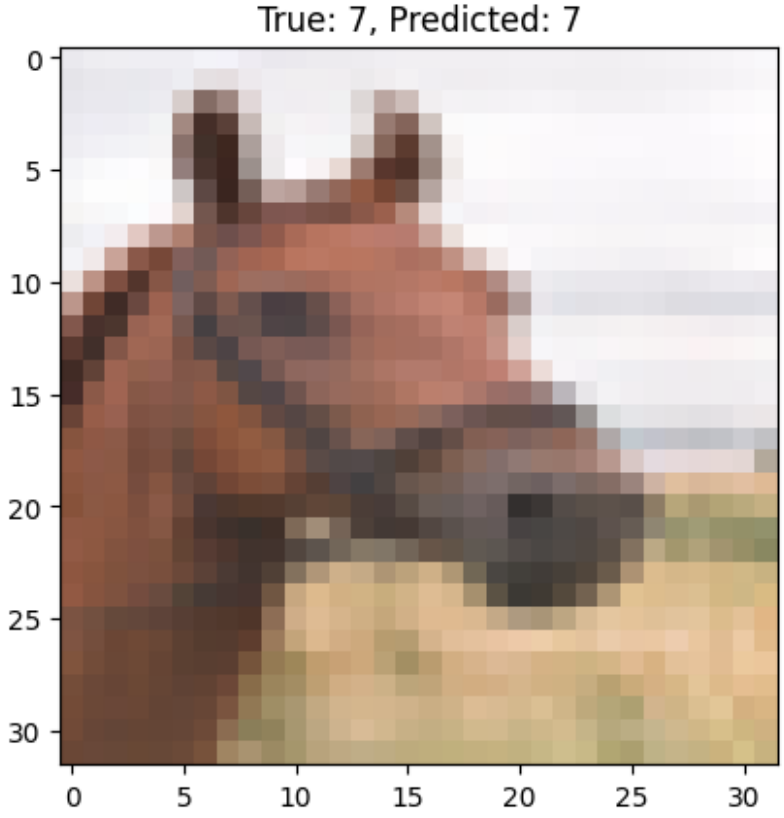

True: 4, Predicted: 4
0
5
10
15
20
25
30
0
5
10
15
20
25
30


True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30


True: 6, Predicted: 6
0
5
10
15
20
25
30
0
5
10
15
20
25
30

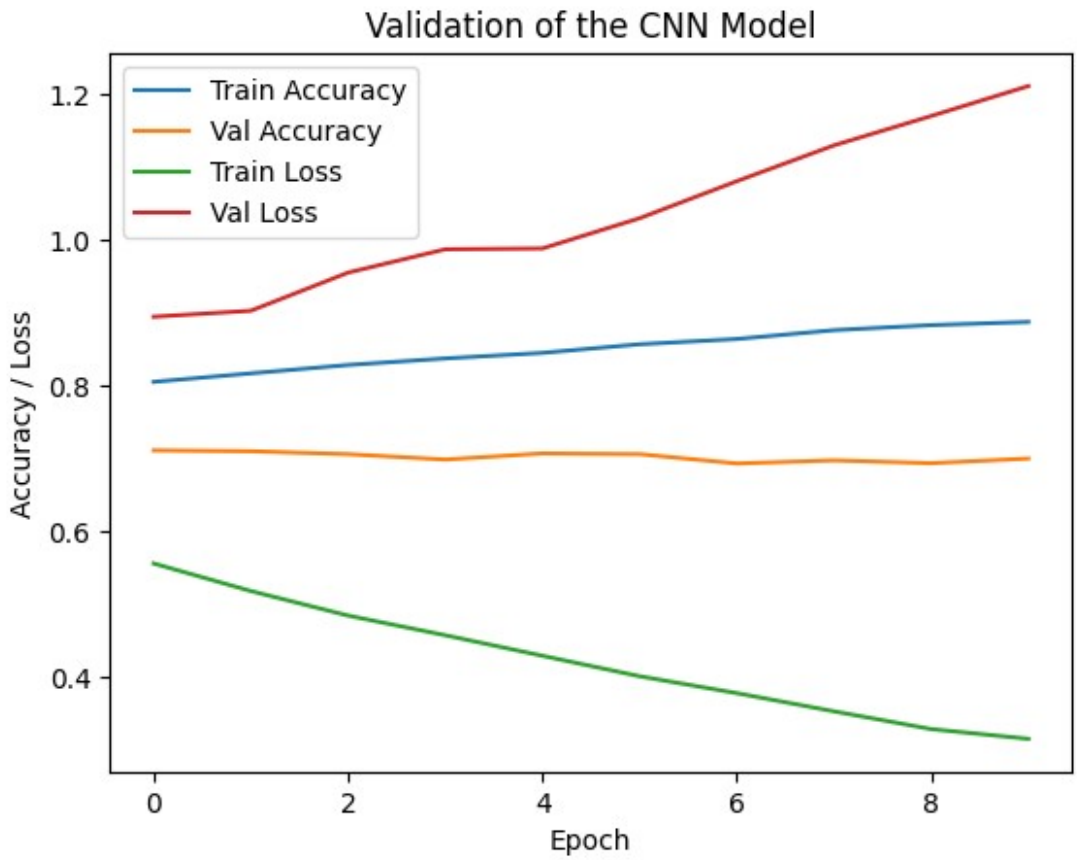

True: 0, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30


True: 0, Predicted: 0
0
5
10
15
20
25
30
0
5
10
15
20
25
30


True: 7, Predicted: 7
0
5
10
15
20
25
30
0
5
10
15
20
25
30

```
#-----------------------------
VALIDATION-----------------------------------------------------------------
--

history = model.fit(train_images, train_labels,
epochs=10,validation_data=(test_images, test_labels))
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.xlabel('Epoch')
plt.ylabel('Accuracy / Loss')
plt.legend()
plt.title('Validation of the CNN Model')
plt.show()
```

```
Epoch 1/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 71s 45ms/step - accuracy: 0.8117 -
loss: 0.5408 - val_accuracy: 0.7110 - val_loss: 0.8944
Epoch 2/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 85s 47ms/step - accuracy: 0.8247 -
loss: 0.4916 - val_accuracy: 0.7098 - val_loss: 0.9026
Epoch 3/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 79s 45ms/step - accuracy: 0.8356 -
loss: 0.4644 - val_accuracy: 0.7058 - val_loss: 0.9549
Epoch 4/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 73s 46ms/step - accuracy: 0.8455 -
loss: 0.4358 - val_accuracy: 0.6984 - val_loss: 0.9870
Epoch 5/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 80s 45ms/step - accuracy: 0.8532 -
loss: 0.4086 - val_accuracy: 0.7068 - val_loss: 0.9881
Epoch 6/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 82s 45ms/step - accuracy: 0.8653 -
loss: 0.3748 - val_accuracy: 0.7058 - val_loss: 1.0296
Epoch 7/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 69s 44ms/step - accuracy: 0.8738 -
loss: 0.3505 - val_accuracy: 0.6930 - val_loss: 1.0804
Epoch 8/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 84s 45ms/step - accuracy: 0.8862 -
loss: 0.3254 - val_accuracy: 0.6973 - val_loss: 1.1296
Epoch 9/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 71s 46ms/step - accuracy: 0.8936 -
loss: 0.3030 - val_accuracy: 0.6934 - val_loss: 1.1698
Epoch 10/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 71s 45ms/step - accuracy: 0.8957 -
loss: 0.2952 - val_accuracy: 0.6998 - val_loss: 1.2110
```




```
#--------------------------VALIDATE THE
IMAGES-----------------------------------------

for i in random.sample(range(len(test_images)), 4):
    plt.imshow(test_images[i])
    plt.title(f"True: {y_true[i]}, Pred: {y_pred_classes[i]}",
color='green' if y_true[i]==y_pred_classes[i] else 'red')
    plt.axis('off')
    plt.show()
```

True: 6, Pred: 6

True: 0, Pred: 0

True: 1, Pred: 1

True: 4, Pred: 4

```
#-------------------ACCURACY PER CLASS
LABEL-------------------------------------------

class_correct = defaultdict(int)
class_total = defaultdict(int)

for true, pred in zip(y_true, y_pred_classes):
    class_total[true] += 1
    if true == pred:
        class_correct[true] += 1

class_accuracy = {k: class_correct[k] / class_total[k] for k in
class_total}
plt.bar(class_accuracy.keys(), class_accuracy.values())
plt.xlabel('Class Index')
plt.ylabel('Accuracy')
plt.title('Per-Class Label Accuracy')
plt.show()
```