In [1]:
```
import pprint
import rasterio
from rasterio import features
import rasterio.warp
import matplotlib.pyplot as plt
import numpy as np
from scipy.ndimage import zoom
import os

import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
import torchvision
from torchvision import datasets, transforms
from torch.utils.data import Dataset
from torch.optim.lr_scheduler import StepLR
```

In [2]:
```
# Base values
channel_descriptions = ['M11', 'I2', 'I1', 'NDVI_last', 'EVI2_last', 'total precipi

min_values = [np.float32(-100.0),
  np.float32(-100.0),
  np.float32(-100.0),
  np.float32(-9863.268),
  np.float32(-4422.217),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(-84.0),
  np.float32(-6.72),
  np.float32(1.0),
  np.float32(0.0),
  np.float32(0.0),
  np.float32(-89.999214),
  np.float32(-13.984883),
  np.float32(0.0),
  np.float32(0.0)]

max_values = [np.float32(15976.0),
  np.float32(15799.0),
  np.float32(15744.0),
  np.float32(9975.073),
  np.float32(9856.787),
  np.float32(122.0),
  np.float32(16.2),
  np.float32(360.0),
  np.float32(311.8),
```

```
 np.float32(325.4),
 np.float32(122.0),
 np.float32(0.01888),
 np.float32(63.85685),
 np.float32(359.42383),
 np.float32(4268.336),
 np.float32(8.28),
 np.float32(17.0),
 np.float32(204.1875),
 np.float32(14.295916),
 np.float32(89.98897),
 np.float32(39.505894),
 np.float32(0.0122514665),
 np.float32(2218.0)]
```

In [3]:

```
fire_folders = []
look_back = 5   # 5 days sequence
all_frames = []
data_limit, loop_counter, loop_start = 30, 0, 0
channel_descriptions = None
base_path = "./data"

target_shape_h, target_shape_w = 128, 128

print('Loading...')

for fire_folder in os.listdir(base_path):
    loop_counter += 1
    if loop_counter - loop_start > data_limit:
        break
    if loop_counter < loop_start:
        continue
    print('Progress: {0}/{1} ({2})'.format(loop_counter - loop_start, data_limit, l
    fire_folders.append(fire_folder)
    for image_name in os.listdir(base_path + f"/{fire_folder}"):
        file_path = base_path + f"/{fire_folder}/{image_name}"

        with rasterio.open(file_path, 'r') as geotiff:
            src = geotiff.read()
            channel_descriptions = geotiff.descriptions
            zoom_factor = (1, target_shape_h / src.shape[1], target_shape_w / src.s
            resized_src = zoom(src, zoom_factor, order=1)
            resized_src = np.nan_to_num(resized_src, copy=True)
            all_frames.append(resized_src)

print(f'Loading done! Count = {len(all_frames)} | Shape = {all_frames[0].shape}')

data_frames = np.stack(all_frames)
print(data_frames.shape)
```

```
Loading...
Progress: 1/30 (1)
Progress: 2/30 (2)
Progress: 3/30 (3)
Progress: 4/30 (4)
Progress: 5/30 (5)
Progress: 6/30 (6)
Progress: 7/30 (7)
Progress: 8/30 (8)
Progress: 9/30 (9)
Progress: 10/30 (10)
Progress: 11/30 (11)
Progress: 12/30 (12)
Progress: 13/30 (13)
Progress: 14/30 (14)
Progress: 15/30 (15)
Progress: 16/30 (16)
Progress: 17/30 (17)
Progress: 18/30 (18)
Progress: 19/30 (19)
Progress: 20/30 (20)
Progress: 21/30 (21)
Progress: 22/30 (22)
Progress: 23/30 (23)
Progress: 24/30 (24)
Progress: 25/30 (25)
Progress: 26/30 (26)
Progress: 27/30 (27)
Progress: 28/30 (28)
Progress: 29/30 (29)
Progress: 30/30 (30)
Loading done! Count = 726 | Shape = (23, 128, 128)
(726, 23, 128, 128)
```

In [4]:

```
from sklearn.preprocessing import MinMaxScaler, minmax_scale

#data_frames = (data_frames - min_bound) / (max_bound - min_bound)

for c in range(23):
    data_frames[:, c, :, :] = (data_frames[:, c, :, :] - min_values[c]) / (max_valu
```

In [5]:

```
np.min(data_frames), np.max(data_frames)
```

Out[5]:

```
(np.float32(0.0), np.float32(1.0))
```

In [6]:

```
X = []
Y = []

for t in range(0, data_frames.shape[0] - look_back):
    x_seq = data_frames[t:t+look_back]               # shape: (5, 23, H, W)
    y_mask = data_frames[t + look_back, 22]           # fire mask from channel 22

    X.append(x_seq)
    Y.append(y_mask)     # binarize
```

```
X = np.stack(X)  # shape: (273, 5, 23, 128, 128)
Y = np.expand_dims(np.stack(Y), axis=1)  # shape: (273, 1, 128, 128)

print(X.shape)
print(Y.shape)
```

```
(721, 5, 23, 128, 128)
(721, 1, 128, 128)
```

In [7]:
```
split_index = int(X.shape[0] * 0.8)

X_train = X[:split_index]
X_val = X[split_index:]

Y_train = Y[:split_index]
Y_val = Y[split_index:]

X_train.shape, Y_train.shape, X_val.shape, Y_val.shape
```

Out[7]:
```
((576, 5, 23, 128, 128),
 (576, 1, 128, 128),
 (145, 5, 23, 128, 128),
 (145, 1, 128, 128))
```

In [8]:
```
X_train = torch.tensor(X_train).float()
Y_train = torch.tensor(Y_train).float()

X_val = torch.tensor(X_val).float()
Y_val = torch.tensor(Y_val).float()

X_train.shape, Y_train.shape, X_val.shape, Y_val.shape
```

Out[8]:
```
(torch.Size([576, 5, 23, 128, 128]),
 torch.Size([576, 1, 128, 128]),
 torch.Size([145, 5, 23, 128, 128]),
 torch.Size([145, 1, 128, 128]))
```

In [9]:
```
class WildfireDataset(Dataset):
    def __init__(self, X, Y):
        self.X = X
        self.Y = Y

    def __len__(self):
        return len(self.X)

    def __getitem__(self, i):
        return self.X[i], self.Y[i]

train_dataset = WildfireDataset(X_train, Y_train)
val_dataset = WildfireDataset(X_val, Y_val)

len(train_dataset), len(val_dataset)
```

Out[9]:
```
(576, 145)
```

In [10]:
```
from torch.utils.data import DataLoader

batch_size = 8

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=False)
val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False)
```

In [11]:
```
if torch.cuda.is_available():
    device = torch.device("cuda")
elif torch.mps.is_available():
    device = torch.device("mps")
else:
    device = torch.device("cpu")

device
```

Out[11]:
```
device(type='cuda')
```

In [12]:
```
import convlstm

class Net(nn.Module):
    def __init__(self):
        super().__init__()
        orig_size = (330, 257)

        self.clstm = convlstm.ConvLSTM(
            input_size=(128, 128),
            input_dim=23,
            hidden_dim=[64],
            kernel_size=(3, 3),
            num_layers=1
        )
        # (8, 64, 128, 128)

        self.head = nn.Sequential(
            nn.Conv2d(64, 1, kernel_size=3, padding=1),
            nn.Sigmoid()
        )
        # (8, 1, 128, 128)

    def forward(self, x):
        batch_size = x.size(0)
        outputs, last_states = self.clstm(x)

        x = outputs[0][:, -1, :, :, :]
        x = self.head(x)
        return x

model = Net().to(device)
```

In [13]:
```
class DiceLoss(nn.Module):
    def __init__(self, smooth=1.0):
        super().__init__()
        self.smooth = smooth
```

```
    def forward(self, y_pred, y_true):
        # (8, 1, 128, 128)

        y_pred_flat = y_pred.view(y_pred.size(0), -1) # (8, 16384)
        y_true_flat = y_true.view(y_true.size(0), -1) # (8, 16384)

        intersection = (y_pred_flat * y_true_flat).sum(dim=1)
        union = y_pred_flat.sum(dim=1) + y_true_flat.sum(dim=1)

        dice_score = (2 * intersection + self.smooth) / (union + self.smooth)
        dice_loss = 1 - dice_score.mean()

        return dice_loss

class BCEWeightedLoss(nn.Module):
    def __init__(self, eps=1e-7):
        super(BCEWeightedLoss, self).__init__()
        self.eps = eps  # to avoid log(0)
        self.pos_weight = 10.0
        self.neg_weight = 1.0
        self.dice = DiceLoss()

    def forward(self, y_pred, y_true):
        """
        y_pred: probabilities after sigmoid, shape (B, 1, H, W)
        y_true: binary targets, shape (B, 1, H, W)
        """
        # Clamp predictions to avoid log(0)
        y_pred = torch.clamp(y_pred, self.eps, 1.0 - self.eps)

        # BCE loss calculation
        loss = - (self.pos_weight * y_true * torch.log(y_pred) + self.neg_weight *

        return loss.mean()  # return scalar loss
```

In [14]:
```
lr = 0.001 # 0.0001
num_epochs = 100

loss_fn = BCEWeightedLoss()

optimizer = torch.optim.Adam(model.parameters(), lr=lr)
print_batch_every = 8

def train(epoch):
    model.train(True)
    print(f"Epoch: {epoch + 1}")
    running_loss = 0.0

    for batch_index, batch in enumerate(train_loader):
        x_batch, y_batch = batch[0].to(device), batch[1].to(device)
        y_pred = model(x_batch)

        loss = loss_fn(y_pred, y_batch)
        running_loss += loss.item()
```

```
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if batch_index % print_batch_every == (print_batch_every - 1):
            avg_loss_across_batches = running_loss / print_batch_every
            print('Batch {0}, Loss: {1:.3f}'.format(batch_index + 1, avg_loss_acros
            running_loss = 0.0

    print()

def validate():
    model.train(False)
    running_loss = 0.0

    for batch_index, batch in enumerate(val_loader):
        x_batch, y_batch = batch[0].to(device), batch[1].to(device)

        with torch.no_grad():
            y_pred = model(x_batch)
            loss = loss_fn(y_pred, y_batch)
            running_loss += loss.item()

    avg_loss_across_batches = running_loss / len(val_loader)
    print('Val Loss: {0:.3f}'.format(avg_loss_across_batches))
    print('***********************************************************')
    print()


for epoch in range(num_epochs):
    train(epoch)
    validate()
```

```
Epoch: 1
```

```
C:\Users\neelr\AppData\Local\Temp\ipykernel_29496\3642139523.py:19: UserWarning: Con
verting a tensor with requires_grad=True to a scalar may lead to unexpected behavio
r.
Consider using tensor.detach() first. (Triggered internally at C:\actions-runner\_wo
rk\pytorch\pytorch\pytorch\aten\src\ATen\native\Scalar.cpp:23.)
  running_loss += loss.item()
```

```
Batch 8, Loss: 0.261
Batch 16, Loss: 0.043
Batch 24, Loss: 0.022
Batch 32, Loss: 0.047
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.167
Batch 64, Loss: 0.099
Batch 72, Loss: 0.198

Val Loss: 0.096
**********************************************************

Epoch: 2
Batch 8, Loss: 0.045
Batch 16, Loss: 0.040
Batch 24, Loss: 0.020
Batch 32, Loss: 0.044
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.168
Batch 64, Loss: 0.099
Batch 72, Loss: 0.199

Val Loss: 0.103
**********************************************************

Epoch: 3
Batch 8, Loss: 0.050
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.044
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.173
Batch 64, Loss: 0.100
Batch 72, Loss: 0.201

Val Loss: 0.113
**********************************************************

Epoch: 4
Batch 8, Loss: 0.056
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.043
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.172
Batch 64, Loss: 0.100
Batch 72, Loss: 0.201

Val Loss: 0.113
**********************************************************

Epoch: 5
```

```
Batch 8, Loss: 0.057
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.043
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.173
Batch 64, Loss: 0.101
Batch 72, Loss: 0.202

Val Loss: 0.115
**********************************************************

Epoch: 6
Batch 8, Loss: 0.058
Batch 16, Loss: 0.038
Batch 24, Loss: 0.020
Batch 32, Loss: 0.043
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.174
Batch 64, Loss: 0.102
Batch 72, Loss: 0.203

Val Loss: 0.117
**********************************************************

Epoch: 7
Batch 8, Loss: 0.061
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.043
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.172
Batch 64, Loss: 0.100
Batch 72, Loss: 0.199

Val Loss: 0.096
**********************************************************

Epoch: 8
Batch 8, Loss: 0.042
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.043
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.169
Batch 64, Loss: 0.099
Batch 72, Loss: 0.199

Val Loss: 0.098
**********************************************************

Epoch: 9
```

```
Batch 8, Loss: 0.045
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.043
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.170
Batch 64, Loss: 0.099
Batch 72, Loss: 0.201

Val Loss: 0.107
**********************************************************

Epoch: 10
Batch 8, Loss: 0.047
Batch 16, Loss: 0.040
Batch 24, Loss: 0.020
Batch 32, Loss: 0.044
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.169
Batch 64, Loss: 0.098
Batch 72, Loss: 0.194

Val Loss: 0.094
**********************************************************

Epoch: 11
Batch 8, Loss: 0.048
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.042
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.167
Batch 64, Loss: 0.099
Batch 72, Loss: 0.199

Val Loss: 0.106
**********************************************************

Epoch: 12
Batch 8, Loss: 0.052
Batch 16, Loss: 0.038
Batch 24, Loss: 0.020
Batch 32, Loss: 0.043
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.171
Batch 64, Loss: 0.099
Batch 72, Loss: 0.200

Val Loss: 0.110
**********************************************************

Epoch: 13
```

```
Batch 8, Loss: 0.054
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.044
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.173
Batch 64, Loss: 0.100
Batch 72, Loss: 0.200

Val Loss: 0.113
*********************************************************

Epoch: 14
Batch 8, Loss: 0.053
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.044
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.172
Batch 64, Loss: 0.099
Batch 72, Loss: 0.199

Val Loss: 0.111
*********************************************************

Epoch: 15
Batch 8, Loss: 0.052
Batch 16, Loss: 0.039
Batch 24, Loss: 0.020
Batch 32, Loss: 0.041
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.149
Batch 64, Loss: 0.091
Batch 72, Loss: 0.140

Val Loss: 0.058
*********************************************************

Epoch: 16
Batch 8, Loss: 0.025
Batch 16, Loss: 0.037
Batch 24, Loss: 0.020
Batch 32, Loss: 0.038
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.134
Batch 64, Loss: 0.083
Batch 72, Loss: 0.109

Val Loss: 0.053
*********************************************************

Epoch: 17
```

```
Batch 8, Loss: 0.025
Batch 16, Loss: 0.036
Batch 24, Loss: 0.020
Batch 32, Loss: 0.036
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.107
Batch 64, Loss: 0.061
Batch 72, Loss: 0.109

Val Loss: 0.058
**********************************************************

Epoch: 18
Batch 8, Loss: 0.023
Batch 16, Loss: 0.037
Batch 24, Loss: 0.019
Batch 32, Loss: 0.038
Batch 40, Loss: 0.015
Batch 48, Loss: 0.008
Batch 56, Loss: 0.133
Batch 64, Loss: 0.068
Batch 72, Loss: 0.100

Val Loss: 0.052
**********************************************************

Epoch: 19
Batch 8, Loss: 0.023
Batch 16, Loss: 0.029
Batch 24, Loss: 0.012
Batch 32, Loss: 0.031
Batch 40, Loss: 0.014
Batch 48, Loss: 0.007
Batch 56, Loss: 0.099
Batch 64, Loss: 0.061
Batch 72, Loss: 0.104

Val Loss: 0.057
**********************************************************

Epoch: 20
Batch 8, Loss: 0.024
Batch 16, Loss: 0.028
Batch 24, Loss: 0.012
Batch 32, Loss: 0.030
Batch 40, Loss: 0.014
Batch 48, Loss: 0.007
Batch 56, Loss: 0.100
Batch 64, Loss: 0.057
Batch 72, Loss: 0.085

Val Loss: 0.050
**********************************************************

Epoch: 21
```

```
Batch 8, Loss: 0.019
Batch 16, Loss: 0.026
Batch 24, Loss: 0.011
Batch 32, Loss: 0.027
Batch 40, Loss: 0.014
Batch 48, Loss: 0.007
Batch 56, Loss: 0.094
Batch 64, Loss: 0.055
Batch 72, Loss: 0.085

Val Loss: 0.050
*********************************************************

Epoch: 22
Batch 8, Loss: 0.019
Batch 16, Loss: 0.026
Batch 24, Loss: 0.011
Batch 32, Loss: 0.026
Batch 40, Loss: 0.014
Batch 48, Loss: 0.007
Batch 56, Loss: 0.094
Batch 64, Loss: 0.068
Batch 72, Loss: 0.118

Val Loss: 0.058
*********************************************************

Epoch: 23
Batch 8, Loss: 0.030
Batch 16, Loss: 0.028
Batch 24, Loss: 0.011
Batch 32, Loss: 0.030
Batch 40, Loss: 0.013
Batch 48, Loss: 0.008
Batch 56, Loss: 0.088
Batch 64, Loss: 0.056
Batch 72, Loss: 0.084

Val Loss: 0.050
*********************************************************

Epoch: 24
Batch 8, Loss: 0.018
Batch 16, Loss: 0.025
Batch 24, Loss: 0.010
Batch 32, Loss: 0.026
Batch 40, Loss: 0.013
Batch 48, Loss: 0.007
Batch 56, Loss: 0.087
Batch 64, Loss: 0.054
Batch 72, Loss: 0.082

Val Loss: 0.049
*********************************************************

Epoch: 25
```

```
Batch 8, Loss: 0.018
Batch 16, Loss: 0.024
Batch 24, Loss: 0.010
Batch 32, Loss: 0.026
Batch 40, Loss: 0.013
Batch 48, Loss: 0.008
Batch 56, Loss: 0.078
Batch 64, Loss: 0.053
Batch 72, Loss: 0.082

Val Loss: 0.048
**********************************************************

Epoch: 26
Batch 8, Loss: 0.016
Batch 16, Loss: 0.025
Batch 24, Loss: 0.010
Batch 32, Loss: 0.026
Batch 40, Loss: 0.013
Batch 48, Loss: 0.008
Batch 56, Loss: 0.078
Batch 64, Loss: 0.052
Batch 72, Loss: 0.079

Val Loss: 0.048
**********************************************************

Epoch: 27
Batch 8, Loss: 0.015
Batch 16, Loss: 0.023
Batch 24, Loss: 0.009
Batch 32, Loss: 0.024
Batch 40, Loss: 0.013
Batch 48, Loss: 0.007
Batch 56, Loss: 0.072
Batch 64, Loss: 0.052
Batch 72, Loss: 0.080

Val Loss: 0.048
**********************************************************

Epoch: 28
Batch 8, Loss: 0.016
Batch 16, Loss: 0.025
Batch 24, Loss: 0.010
Batch 32, Loss: 0.026
Batch 40, Loss: 0.013
Batch 48, Loss: 0.007
Batch 56, Loss: 0.077
Batch 64, Loss: 0.051
Batch 72, Loss: 0.075

Val Loss: 0.048
**********************************************************

Epoch: 29
```

```
Batch 8, Loss: 0.014
Batch 16, Loss: 0.022
Batch 24, Loss: 0.009
Batch 32, Loss: 0.023
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.068
Batch 64, Loss: 0.049
Batch 72, Loss: 0.076

Val Loss: 0.049
**********************************************************

Epoch: 30
Batch 8, Loss: 0.016
Batch 16, Loss: 0.025
Batch 24, Loss: 0.010
Batch 32, Loss: 0.025
Batch 40, Loss: 0.013
Batch 48, Loss: 0.007
Batch 56, Loss: 0.073
Batch 64, Loss: 0.049
Batch 72, Loss: 0.074

Val Loss: 0.048
**********************************************************

Epoch: 31
Batch 8, Loss: 0.014
Batch 16, Loss: 0.022
Batch 24, Loss: 0.009
Batch 32, Loss: 0.023
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.066
Batch 64, Loss: 0.048
Batch 72, Loss: 0.074

Val Loss: 0.047
**********************************************************

Epoch: 32
Batch 8, Loss: 0.014
Batch 16, Loss: 0.023
Batch 24, Loss: 0.010
Batch 32, Loss: 0.022
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.064
Batch 64, Loss: 0.047
Batch 72, Loss: 0.073

Val Loss: 0.048
**********************************************************

Epoch: 33
```

```
Batch 8, Loss: 0.015
Batch 16, Loss: 0.024
Batch 24, Loss: 0.011
Batch 32, Loss: 0.024
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.068
Batch 64, Loss: 0.047
Batch 72, Loss: 0.072

Val Loss: 0.046
**********************************************************

Epoch: 34
Batch 8, Loss: 0.015
Batch 16, Loss: 0.024
Batch 24, Loss: 0.009
Batch 32, Loss: 0.022
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.063
Batch 64, Loss: 0.046
Batch 72, Loss: 0.071

Val Loss: 0.046
**********************************************************

Epoch: 35
Batch 8, Loss: 0.014
Batch 16, Loss: 0.024
Batch 24, Loss: 0.009
Batch 32, Loss: 0.024
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.063
Batch 64, Loss: 0.047
Batch 72, Loss: 0.072

Val Loss: 0.046
**********************************************************

Epoch: 36
Batch 8, Loss: 0.014
Batch 16, Loss: 0.023
Batch 24, Loss: 0.009
Batch 32, Loss: 0.022
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.060
Batch 64, Loss: 0.046
Batch 72, Loss: 0.071

Val Loss: 0.046
**********************************************************

Epoch: 37
```

```
Batch 8, Loss: 0.013
Batch 16, Loss: 0.023
Batch 24, Loss: 0.009
Batch 32, Loss: 0.022
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.062
Batch 64, Loss: 0.045
Batch 72, Loss: 0.071

Val Loss: 0.044
**********************************************************

Epoch: 38
Batch 8, Loss: 0.014
Batch 16, Loss: 0.023
Batch 24, Loss: 0.009
Batch 32, Loss: 0.023
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.058
Batch 64, Loss: 0.045
Batch 72, Loss: 0.068

Val Loss: 0.043
**********************************************************

Epoch: 39
Batch 8, Loss: 0.013
Batch 16, Loss: 0.023
Batch 24, Loss: 0.008
Batch 32, Loss: 0.023
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.057
Batch 64, Loss: 0.046
Batch 72, Loss: 0.070

Val Loss: 0.045
**********************************************************

Epoch: 40
Batch 8, Loss: 0.014
Batch 16, Loss: 0.021
Batch 24, Loss: 0.008
Batch 32, Loss: 0.022
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.056
Batch 64, Loss: 0.044
Batch 72, Loss: 0.067

Val Loss: 0.042
**********************************************************

Epoch: 41
```

```
Batch 8, Loss: 0.013
Batch 16, Loss: 0.020
Batch 24, Loss: 0.009
Batch 32, Loss: 0.022
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.055
Batch 64, Loss: 0.045
Batch 72, Loss: 0.068

Val Loss: 0.042
*********************************************************

Epoch: 42
Batch 8, Loss: 0.014
Batch 16, Loss: 0.020
Batch 24, Loss: 0.008
Batch 32, Loss: 0.021
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.053
Batch 64, Loss: 0.045
Batch 72, Loss: 0.065

Val Loss: 0.042
*********************************************************

Epoch: 43
Batch 8, Loss: 0.012
Batch 16, Loss: 0.020
Batch 24, Loss: 0.008
Batch 32, Loss: 0.021
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.053
Batch 64, Loss: 0.044
Batch 72, Loss: 0.065

Val Loss: 0.041
*********************************************************

Epoch: 44
Batch 8, Loss: 0.012
Batch 16, Loss: 0.020
Batch 24, Loss: 0.008
Batch 32, Loss: 0.022
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.055
Batch 64, Loss: 0.045
Batch 72, Loss: 0.068

Val Loss: 0.042
*********************************************************

Epoch: 45
```

```
Batch 8, Loss: 0.013
Batch 16, Loss: 0.020
Batch 24, Loss: 0.008
Batch 32, Loss: 0.021
Batch 40, Loss: 0.012
Batch 48, Loss: 0.007
Batch 56, Loss: 0.055
Batch 64, Loss: 0.042
Batch 72, Loss: 0.066

Val Loss: 0.041
**********************************************************

Epoch: 46
Batch 8, Loss: 0.013
Batch 16, Loss: 0.018
Batch 24, Loss: 0.009
Batch 32, Loss: 0.024
Batch 40, Loss: 0.011
Batch 48, Loss: 0.006
Batch 56, Loss: 0.053
Batch 64, Loss: 0.044
Batch 72, Loss: 0.068

Val Loss: 0.041
**********************************************************

Epoch: 47
Batch 8, Loss: 0.014
Batch 16, Loss: 0.021
Batch 24, Loss: 0.008
Batch 32, Loss: 0.020
Batch 40, Loss: 0.011
Batch 48, Loss: 0.006
Batch 56, Loss: 0.052
Batch 64, Loss: 0.041
Batch 72, Loss: 0.063

Val Loss: 0.039
**********************************************************

Epoch: 48
Batch 8, Loss: 0.012
Batch 16, Loss: 0.019
Batch 24, Loss: 0.009
Batch 32, Loss: 0.021
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.047
Batch 64, Loss: 0.043
Batch 72, Loss: 0.062

Val Loss: 0.039
**********************************************************

Epoch: 49
```

```
Batch 8, Loss: 0.012
Batch 16, Loss: 0.019
Batch 24, Loss: 0.008
Batch 32, Loss: 0.020
Batch 40, Loss: 0.011
Batch 48, Loss: 0.006
Batch 56, Loss: 0.047
Batch 64, Loss: 0.041
Batch 72, Loss: 0.062

Val Loss: 0.038
**********************************************************

Epoch: 50
Batch 8, Loss: 0.012
Batch 16, Loss: 0.018
Batch 24, Loss: 0.008
Batch 32, Loss: 0.021
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.045
Batch 64, Loss: 0.044
Batch 72, Loss: 0.065

Val Loss: 0.040
**********************************************************

Epoch: 51
Batch 8, Loss: 0.013
Batch 16, Loss: 0.020
Batch 24, Loss: 0.008
Batch 32, Loss: 0.020
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.049
Batch 64, Loss: 0.042
Batch 72, Loss: 0.069

Val Loss: 0.038
**********************************************************

Epoch: 52
Batch 8, Loss: 0.013
Batch 16, Loss: 0.019
Batch 24, Loss: 0.009
Batch 32, Loss: 0.020
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.050
Batch 64, Loss: 0.041
Batch 72, Loss: 0.067

Val Loss: 0.040
**********************************************************

Epoch: 53
```

```
Batch 8, Loss: 0.013
Batch 16, Loss: 0.019
Batch 24, Loss: 0.008
Batch 32, Loss: 0.020
Batch 40, Loss: 0.011
Batch 48, Loss: 0.007
Batch 56, Loss: 0.049
Batch 64, Loss: 0.042
Batch 72, Loss: 0.060

Val Loss: 0.038
**********************************************************

Epoch: 54
Batch 8, Loss: 0.012
Batch 16, Loss: 0.018
Batch 24, Loss: 0.008
Batch 32, Loss: 0.019
Batch 40, Loss: 0.011
Batch 48, Loss: 0.006
Batch 56, Loss: 0.048
Batch 64, Loss: 0.041
Batch 72, Loss: 0.065

Val Loss: 0.039
**********************************************************

Epoch: 55
Batch 8, Loss: 0.013
Batch 16, Loss: 0.019
Batch 24, Loss: 0.008
Batch 32, Loss: 0.020
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.046
Batch 64, Loss: 0.042
Batch 72, Loss: 0.060

Val Loss: 0.039
**********************************************************

Epoch: 56
Batch 8, Loss: 0.012
Batch 16, Loss: 0.018
Batch 24, Loss: 0.008
Batch 32, Loss: 0.019
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.044
Batch 64, Loss: 0.040
Batch 72, Loss: 0.060

Val Loss: 0.038
**********************************************************

Epoch: 57
```

```
Batch 8, Loss: 0.012
Batch 16, Loss: 0.018
Batch 24, Loss: 0.008
Batch 32, Loss: 0.019
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.042
Batch 64, Loss: 0.040
Batch 72, Loss: 0.059

Val Loss: 0.038
**********************************************************

Epoch: 58
Batch 8, Loss: 0.013
Batch 16, Loss: 0.017
Batch 24, Loss: 0.007
Batch 32, Loss: 0.019
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.043
Batch 64, Loss: 0.038
Batch 72, Loss: 0.058

Val Loss: 0.037
**********************************************************

Epoch: 59
Batch 8, Loss: 0.012
Batch 16, Loss: 0.017
Batch 24, Loss: 0.008
Batch 32, Loss: 0.019
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.041
Batch 64, Loss: 0.039
Batch 72, Loss: 0.058

Val Loss: 0.037
**********************************************************

Epoch: 60
Batch 8, Loss: 0.012
Batch 16, Loss: 0.017
Batch 24, Loss: 0.008
Batch 32, Loss: 0.018
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.043
Batch 64, Loss: 0.038
Batch 72, Loss: 0.058

Val Loss: 0.038
**********************************************************

Epoch: 61
```

```
Batch 8, Loss: 0.012
Batch 16, Loss: 0.017
Batch 24, Loss: 0.009
Batch 32, Loss: 0.019
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.045
Batch 64, Loss: 0.040
Batch 72, Loss: 0.058

Val Loss: 0.037
**********************************************************

Epoch: 62
Batch 8, Loss: 0.011
Batch 16, Loss: 0.016
Batch 24, Loss: 0.008
Batch 32, Loss: 0.021
Batch 40, Loss: 0.010
Batch 48, Loss: 0.007
Batch 56, Loss: 0.049
Batch 64, Loss: 0.040
Batch 72, Loss: 0.059

Val Loss: 0.043
**********************************************************

Epoch: 63
Batch 8, Loss: 0.012
Batch 16, Loss: 0.017
Batch 24, Loss: 0.009
Batch 32, Loss: 0.020
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.044
Batch 64, Loss: 0.041
Batch 72, Loss: 0.059

Val Loss: 0.039
**********************************************************

Epoch: 64
Batch 8, Loss: 0.013
Batch 16, Loss: 0.018
Batch 24, Loss: 0.009
Batch 32, Loss: 0.018
Batch 40, Loss: 0.011
Batch 48, Loss: 0.006
Batch 56, Loss: 0.047
Batch 64, Loss: 0.043
Batch 72, Loss: 0.071

Val Loss: 0.038
**********************************************************

Epoch: 65
```

```
Batch 8, Loss: 0.013
Batch 16, Loss: 0.019
Batch 24, Loss: 0.008
Batch 32, Loss: 0.020
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.051
Batch 64, Loss: 0.039
Batch 72, Loss: 0.061

Val Loss: 0.037
**********************************************************

Epoch: 66
Batch 8, Loss: 0.012
Batch 16, Loss: 0.017
Batch 24, Loss: 0.008
Batch 32, Loss: 0.018
Batch 40, Loss: 0.011
Batch 48, Loss: 0.006
Batch 56, Loss: 0.044
Batch 64, Loss: 0.040
Batch 72, Loss: 0.058

Val Loss: 0.038
**********************************************************

Epoch: 67
Batch 8, Loss: 0.012
Batch 16, Loss: 0.016
Batch 24, Loss: 0.008
Batch 32, Loss: 0.018
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.043
Batch 64, Loss: 0.038
Batch 72, Loss: 0.057

Val Loss: 0.038
**********************************************************

Epoch: 68
Batch 8, Loss: 0.012
Batch 16, Loss: 0.016
Batch 24, Loss: 0.008
Batch 32, Loss: 0.019
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.043
Batch 64, Loss: 0.039
Batch 72, Loss: 0.059

Val Loss: 0.040
**********************************************************

Epoch: 69
```

```
Batch 8, Loss: 0.012
Batch 16, Loss: 0.015
Batch 24, Loss: 0.008
Batch 32, Loss: 0.019
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.043
Batch 64, Loss: 0.037
Batch 72, Loss: 0.059

Val Loss: 0.039
**********************************************************

Epoch: 70
Batch 8, Loss: 0.014
Batch 16, Loss: 0.017
Batch 24, Loss: 0.007
Batch 32, Loss: 0.018
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.043
Batch 64, Loss: 0.038
Batch 72, Loss: 0.063

Val Loss: 0.040
**********************************************************

Epoch: 71
Batch 8, Loss: 0.012
Batch 16, Loss: 0.018
Batch 24, Loss: 0.007
Batch 32, Loss: 0.019
Batch 40, Loss: 0.011
Batch 48, Loss: 0.006
Batch 56, Loss: 0.052
Batch 64, Loss: 0.036
Batch 72, Loss: 0.057

Val Loss: 0.042
**********************************************************

Epoch: 72
Batch 8, Loss: 0.013
Batch 16, Loss: 0.016
Batch 24, Loss: 0.008
Batch 32, Loss: 0.019
Batch 40, Loss: 0.010
Batch 48, Loss: 0.005
Batch 56, Loss: 0.046
Batch 64, Loss: 0.036
Batch 72, Loss: 0.058

Val Loss: 0.040
**********************************************************

Epoch: 73
```

```
Batch 8, Loss: 0.013
Batch 16, Loss: 0.017
Batch 24, Loss: 0.007
Batch 32, Loss: 0.018
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.044
Batch 64, Loss: 0.037
Batch 72, Loss: 0.055

Val Loss: 0.041
**********************************************************

Epoch: 74
Batch 8, Loss: 0.011
Batch 16, Loss: 0.015
Batch 24, Loss: 0.008
Batch 32, Loss: 0.018
Batch 40, Loss: 0.010
Batch 48, Loss: 0.006
Batch 56, Loss: 0.043
Batch 64, Loss: 0.036
Batch 72, Loss: 0.053

Val Loss: 0.039
**********************************************************

Epoch: 75
Batch 8, Loss: 0.011
Batch 16, Loss: 0.015
Batch 24, Loss: 0.008
Batch 32, Loss: 0.018
Batch 40, Loss: 0.009
Batch 48, Loss: 0.006
Batch 56, Loss: 0.042
Batch 64, Loss: 0.036
Batch 72, Loss: 0.056

Val Loss: 0.036
**********************************************************

Epoch: 76
Batch 8, Loss: 0.011
Batch 16, Loss: 0.015
Batch 24, Loss: 0.008
Batch 32, Loss: 0.017
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.042
Batch 64, Loss: 0.036
Batch 72, Loss: 0.054

Val Loss: 0.041
**********************************************************

Epoch: 77
```

```
Batch 8, Loss: 0.012
Batch 16, Loss: 0.015
Batch 24, Loss: 0.007
Batch 32, Loss: 0.018
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.041
Batch 64, Loss: 0.035
Batch 72, Loss: 0.053

Val Loss: 0.038
**********************************************************

Epoch: 78
Batch 8, Loss: 0.011
Batch 16, Loss: 0.014
Batch 24, Loss: 0.007
Batch 32, Loss: 0.017
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.039
Batch 64, Loss: 0.035
Batch 72, Loss: 0.053

Val Loss: 0.041
**********************************************************

Epoch: 79
Batch 8, Loss: 0.011
Batch 16, Loss: 0.014
Batch 24, Loss: 0.007
Batch 32, Loss: 0.018
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.038
Batch 64, Loss: 0.035
Batch 72, Loss: 0.054

Val Loss: 0.038
**********************************************************

Epoch: 80
Batch 8, Loss: 0.012
Batch 16, Loss: 0.015
Batch 24, Loss: 0.008
Batch 32, Loss: 0.017
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.038
Batch 64, Loss: 0.034
Batch 72, Loss: 0.053

Val Loss: 0.038
**********************************************************

Epoch: 81
```

```
Batch 8, Loss: 0.011
Batch 16, Loss: 0.015
Batch 24, Loss: 0.007
Batch 32, Loss: 0.017
Batch 40, Loss: 0.010
Batch 48, Loss: 0.005
Batch 56, Loss: 0.040
Batch 64, Loss: 0.034
Batch 72, Loss: 0.054

Val Loss: 0.037
**********************************************************

Epoch: 82
Batch 8, Loss: 0.011
Batch 16, Loss: 0.014
Batch 24, Loss: 0.007
Batch 32, Loss: 0.016
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.035
Batch 64, Loss: 0.032
Batch 72, Loss: 0.053

Val Loss: 0.038
**********************************************************

Epoch: 83
Batch 8, Loss: 0.011
Batch 16, Loss: 0.014
Batch 24, Loss: 0.007
Batch 32, Loss: 0.017
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.038
Batch 64, Loss: 0.033
Batch 72, Loss: 0.049

Val Loss: 0.039
**********************************************************

Epoch: 84
Batch 8, Loss: 0.010
Batch 16, Loss: 0.013
Batch 24, Loss: 0.007
Batch 32, Loss: 0.015
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.036
Batch 64, Loss: 0.033
Batch 72, Loss: 0.052

Val Loss: 0.040
**********************************************************

Epoch: 85
```

```
Batch 8, Loss: 0.011
Batch 16, Loss: 0.014
Batch 24, Loss: 0.008
Batch 32, Loss: 0.017
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.037
Batch 64, Loss: 0.032
Batch 72, Loss: 0.050

Val Loss: 0.041
*********************************************************

Epoch: 86
Batch 8, Loss: 0.011
Batch 16, Loss: 0.014
Batch 24, Loss: 0.008
Batch 32, Loss: 0.016
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.036
Batch 64, Loss: 0.034
Batch 72, Loss: 0.050

Val Loss: 0.041
*********************************************************

Epoch: 87
Batch 8, Loss: 0.010
Batch 16, Loss: 0.013
Batch 24, Loss: 0.008
Batch 32, Loss: 0.015
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.038
Batch 64, Loss: 0.033
Batch 72, Loss: 0.057

Val Loss: 0.043
*********************************************************

Epoch: 88
Batch 8, Loss: 0.011
Batch 16, Loss: 0.014
Batch 24, Loss: 0.007
Batch 32, Loss: 0.015
Batch 40, Loss: 0.010
Batch 48, Loss: 0.005
Batch 56, Loss: 0.043
Batch 64, Loss: 0.035
Batch 72, Loss: 0.051

Val Loss: 0.044
*********************************************************

Epoch: 89
```

```
Batch 8, Loss: 0.010
Batch 16, Loss: 0.014
Batch 24, Loss: 0.008
Batch 32, Loss: 0.015
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.039
Batch 64, Loss: 0.034
Batch 72, Loss: 0.051

Val Loss: 0.045
*********************************************************

Epoch: 90
Batch 8, Loss: 0.010
Batch 16, Loss: 0.013
Batch 24, Loss: 0.008
Batch 32, Loss: 0.015
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.037
Batch 64, Loss: 0.033
Batch 72, Loss: 0.049

Val Loss: 0.041
*********************************************************

Epoch: 91
Batch 8, Loss: 0.010
Batch 16, Loss: 0.012
Batch 24, Loss: 0.008
Batch 32, Loss: 0.015
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.039
Batch 64, Loss: 0.032
Batch 72, Loss: 0.049

Val Loss: 0.040
*********************************************************

Epoch: 92
Batch 8, Loss: 0.010
Batch 16, Loss: 0.012
Batch 24, Loss: 0.008
Batch 32, Loss: 0.015
Batch 40, Loss: 0.008
Batch 48, Loss: 0.004
Batch 56, Loss: 0.034
Batch 64, Loss: 0.031
Batch 72, Loss: 0.049

Val Loss: 0.041
*********************************************************

Epoch: 93
```

```
Batch 8, Loss: 0.010
Batch 16, Loss: 0.012
Batch 24, Loss: 0.007
Batch 32, Loss: 0.015
Batch 40, Loss: 0.008
Batch 48, Loss: 0.004
Batch 56, Loss: 0.034
Batch 64, Loss: 0.031
Batch 72, Loss: 0.049

Val Loss: 0.043
**********************************************************

Epoch: 94
Batch 8, Loss: 0.010
Batch 16, Loss: 0.013
Batch 24, Loss: 0.008
Batch 32, Loss: 0.015
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.033
Batch 64, Loss: 0.031
Batch 72, Loss: 0.046

Val Loss: 0.041
**********************************************************

Epoch: 95
Batch 8, Loss: 0.010
Batch 16, Loss: 0.012
Batch 24, Loss: 0.007
Batch 32, Loss: 0.014
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.032
Batch 64, Loss: 0.029
Batch 72, Loss: 0.045

Val Loss: 0.042
**********************************************************

Epoch: 96
Batch 8, Loss: 0.009
Batch 16, Loss: 0.011
Batch 24, Loss: 0.007
Batch 32, Loss: 0.014
Batch 40, Loss: 0.008
Batch 48, Loss: 0.004
Batch 56, Loss: 0.031
Batch 64, Loss: 0.029
Batch 72, Loss: 0.044

Val Loss: 0.044
**********************************************************

Epoch: 97
```

```
Batch 8, Loss: 0.009
Batch 16, Loss: 0.012
Batch 24, Loss: 0.007
Batch 32, Loss: 0.014
Batch 40, Loss: 0.008
Batch 48, Loss: 0.004
Batch 56, Loss: 0.031
Batch 64, Loss: 0.029
Batch 72, Loss: 0.045

Val Loss: 0.040
*********************************************************

Epoch: 98
Batch 8, Loss: 0.009
Batch 16, Loss: 0.012
Batch 24, Loss: 0.008
Batch 32, Loss: 0.014
Batch 40, Loss: 0.008
Batch 48, Loss: 0.004
Batch 56, Loss: 0.031
Batch 64, Loss: 0.028
Batch 72, Loss: 0.043

Val Loss: 0.043
*********************************************************

Epoch: 99
Batch 8, Loss: 0.009
Batch 16, Loss: 0.011
Batch 24, Loss: 0.006
Batch 32, Loss: 0.013
Batch 40, Loss: 0.008
Batch 48, Loss: 0.004
Batch 56, Loss: 0.031
Batch 64, Loss: 0.032
Batch 72, Loss: 0.053

Val Loss: 0.041
*********************************************************

Epoch: 100
Batch 8, Loss: 0.011
Batch 16, Loss: 0.015
Batch 24, Loss: 0.007
Batch 32, Loss: 0.016
Batch 40, Loss: 0.009
Batch 48, Loss: 0.005
Batch 56, Loss: 0.036
Batch 64, Loss: 0.036
Batch 72, Loss: 0.051

Val Loss: 0.037
*********************************************************
```

In [15]:
```
for batch_index, batch in enumerate(train_loader):
    if batch_index == 4:
        x_batch, y_batch = batch[0].to(device), batch[1].to(device)
        y_pred = model(x_batch)
        print("found")
        break
```

```
found
```

In [16]:
```
torch.max(y_batch), torch.max(y_pred), torch.min(y_batch), torch.min(y_pred)
```

Out[16]:
```
(tensor(0.9495, device='cuda:0'),
 tensor(0.9533, device='cuda:0', grad_fn=<MaxBackward1>),
 tensor(0., device='cuda:0'),
 tensor(2.3402e-06, device='cuda:0', grad_fn=<MinBackward1>))
```

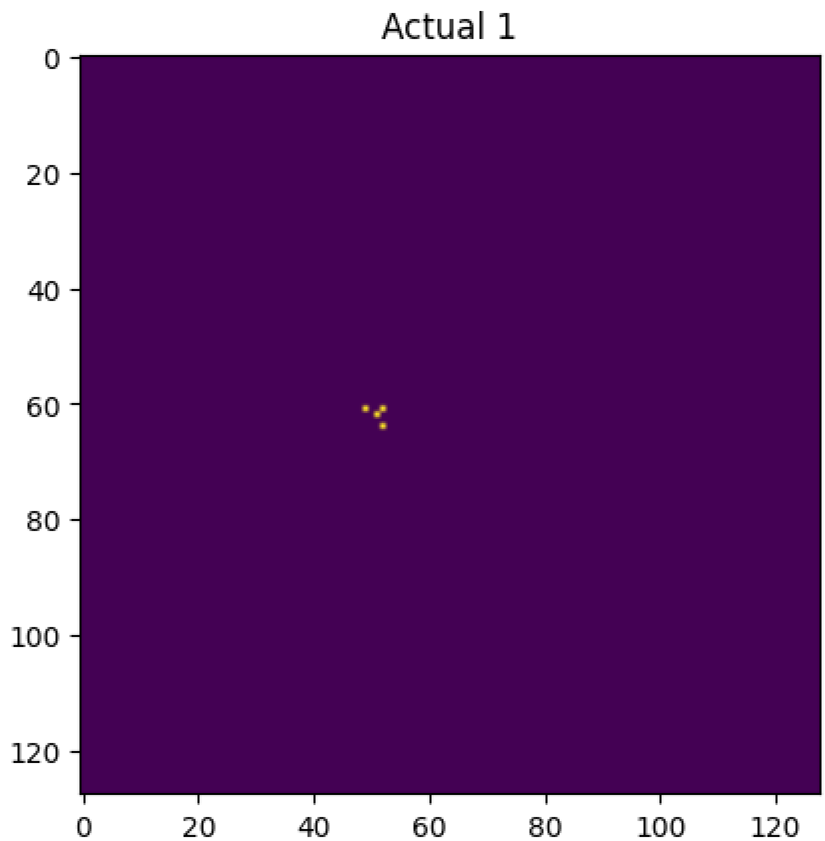
In [17]:
```
y_batch_np = y_batch.cpu().detach().numpy()
y_pred_np = y_pred.cpu().detach().numpy()

y_pred_avg = (np.min(y_pred_np) + np.max(y_pred_np)) / 2.0
#y_pred_np = (y_pred_np - 0.0) / (0.65 - 0.0)

y_batch_np.shape, y_pred_np.shape
```

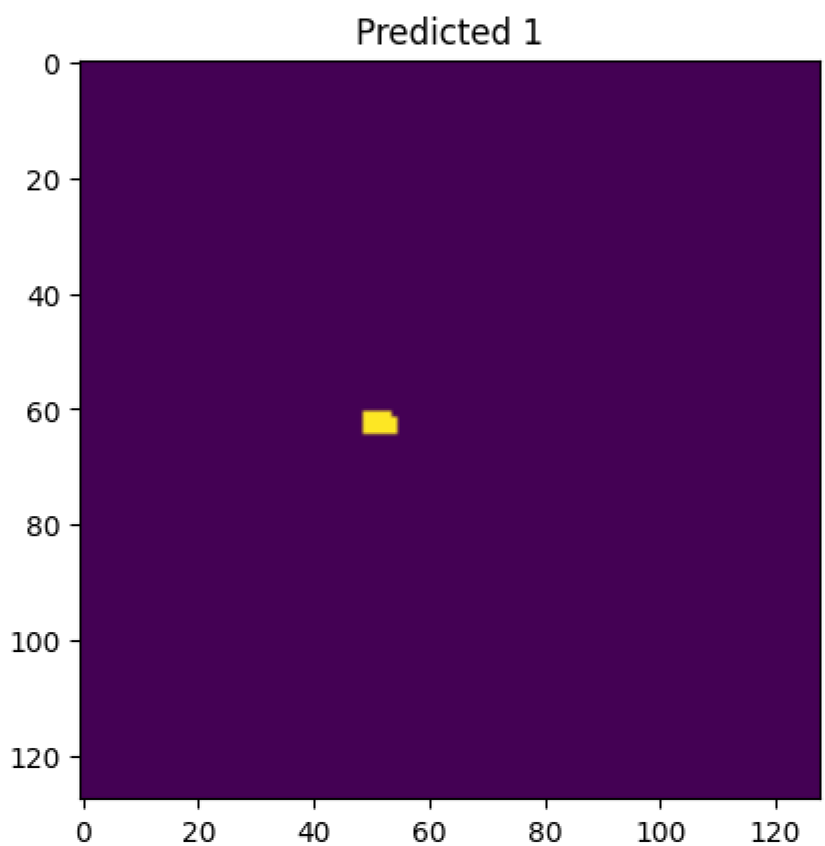
Out[17]:
```
((8, 1, 128, 128), (8, 1, 128, 128))
```

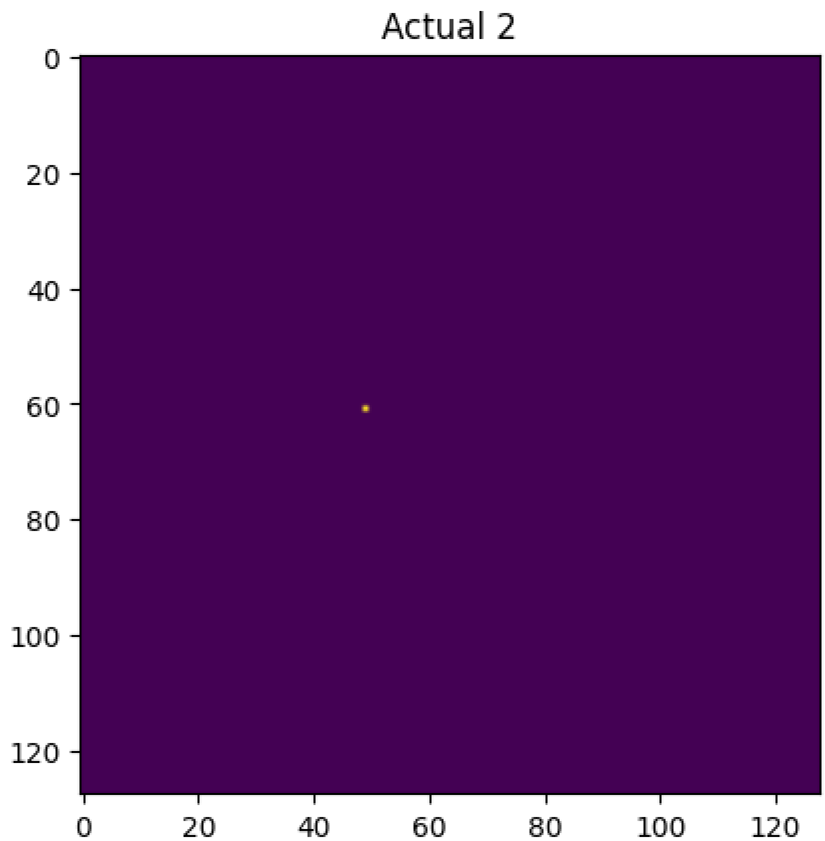
In [18]:
```
np.max(y_pred_np), np.min(y_pred_np), y_pred_avg
```

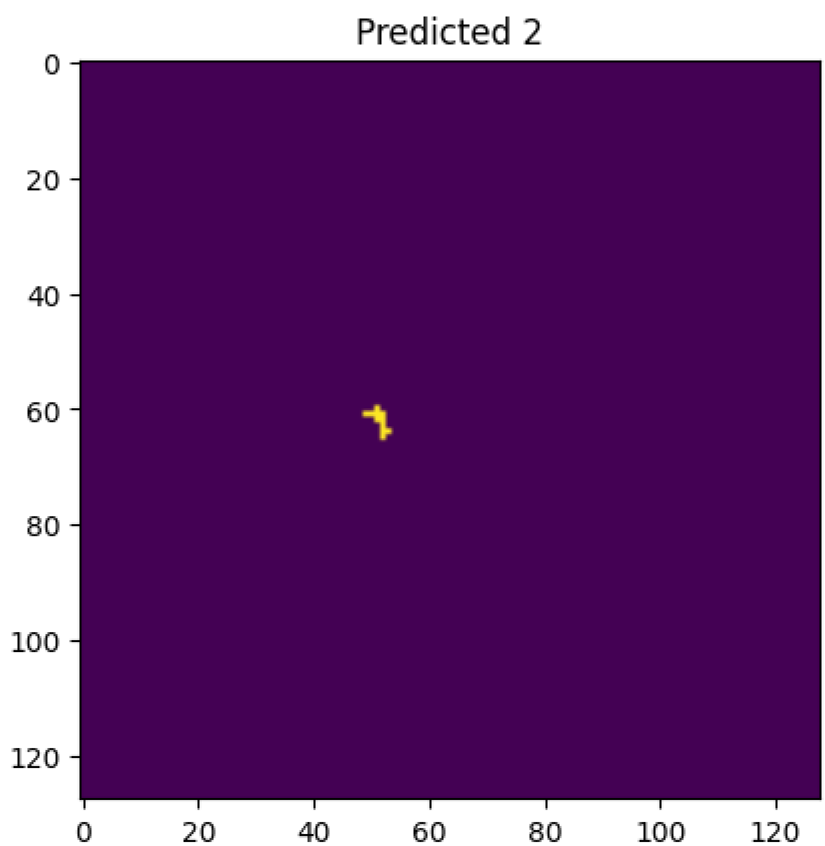
Out[18]:
```
(np.float32(0.9533081), np.float32(2.340184e-06), np.float32(0.47665522))
```

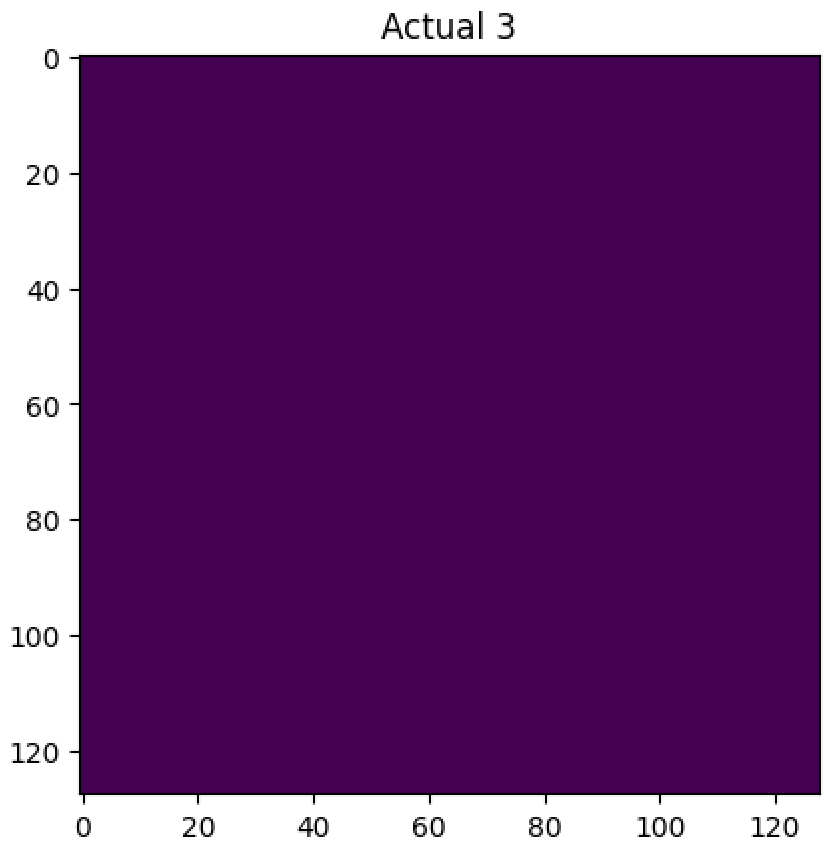
In [19]:
```
for i in range(y_batch_np.shape[0]):
    plt.figure()
    plt.title(f"Actual {i + 1}")
    split_val = 0.15
    val = y_batch_np[i, 0]
    plt.imshow(np.piecewise(val, [val < split_val, val >= split_val], [0, 1]))

    plt.figure()
    plt.title(f"Predicted {i + 1}")
    split_val = 0.5
    val = y_pred_np[i, 0]
    plt.imshow(np.piecewise(val, [val < split_val, val >= split_val], [0, 1]))
    #plt.imshow(val)
```

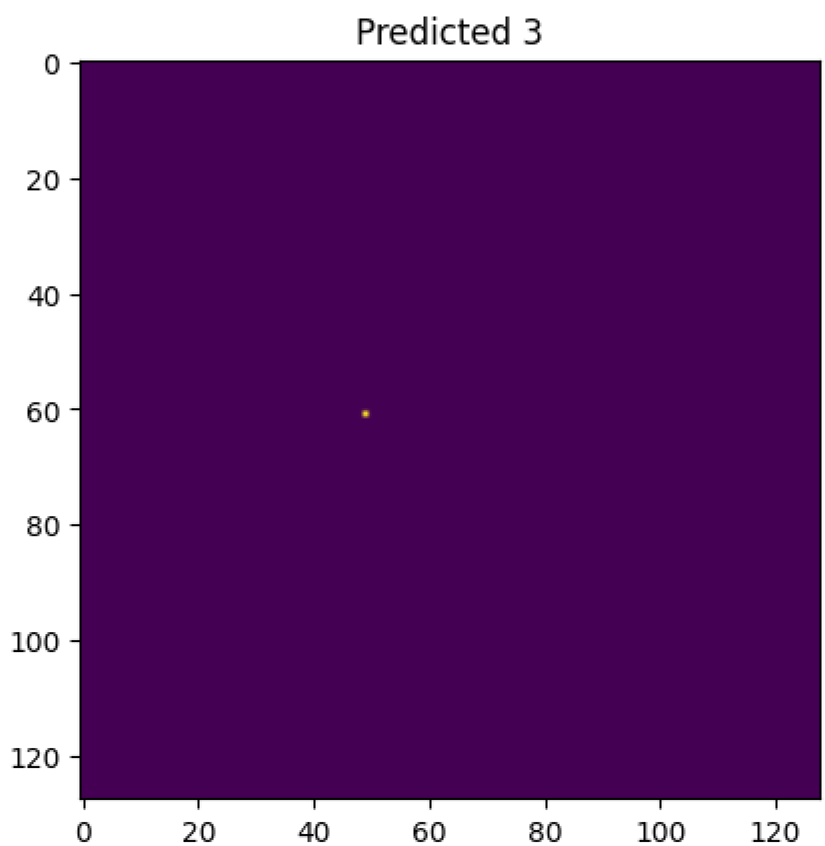

Actual 1
0
20
40
60
80
100
120
0 20 40 60 80 100 120

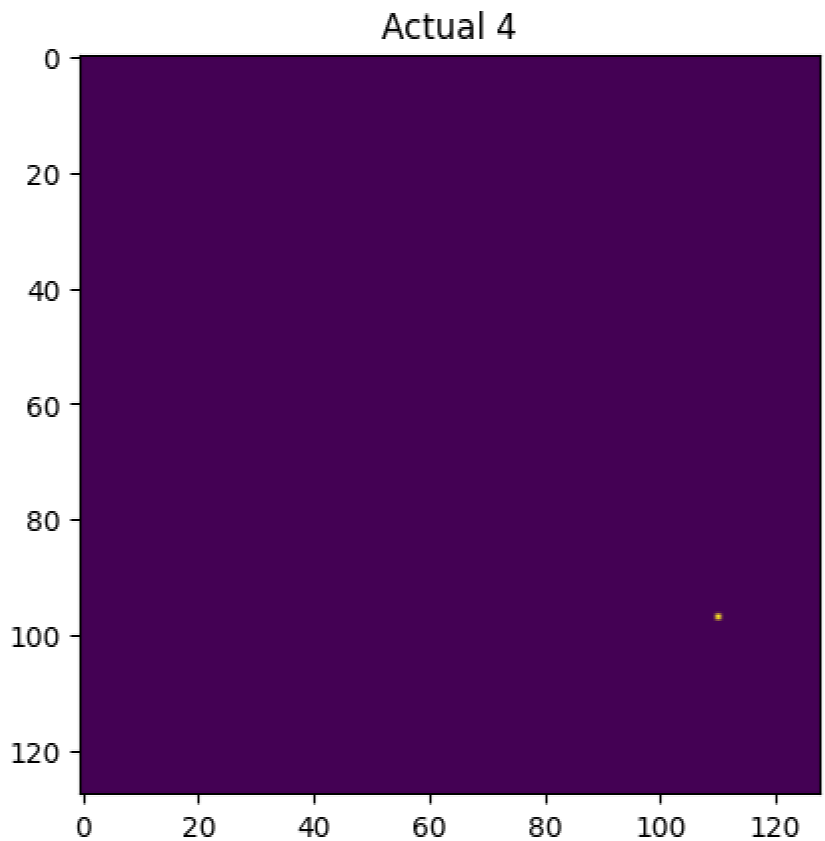

Predicted 1
0
20
40
60
80
100
120
0 20 40 60 80 100 120

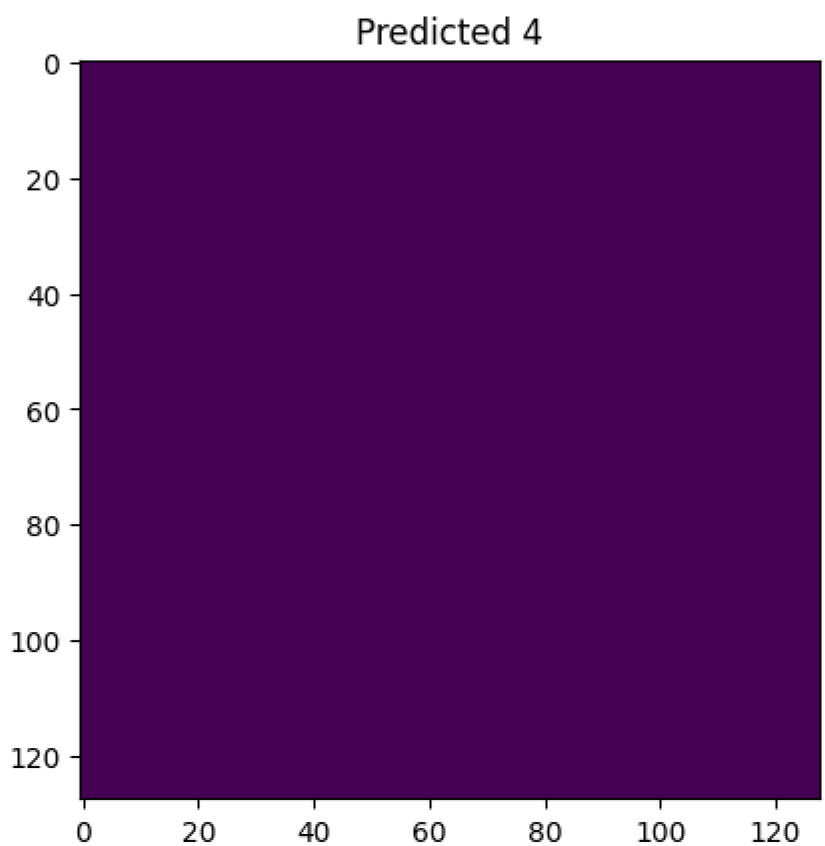

Actual 2
0
20
40
60
80
100
120
0 20 40 60 80 100 120


Predicted 2
0
20
40
60
80
100
120
0 20 40 60 80 100 120

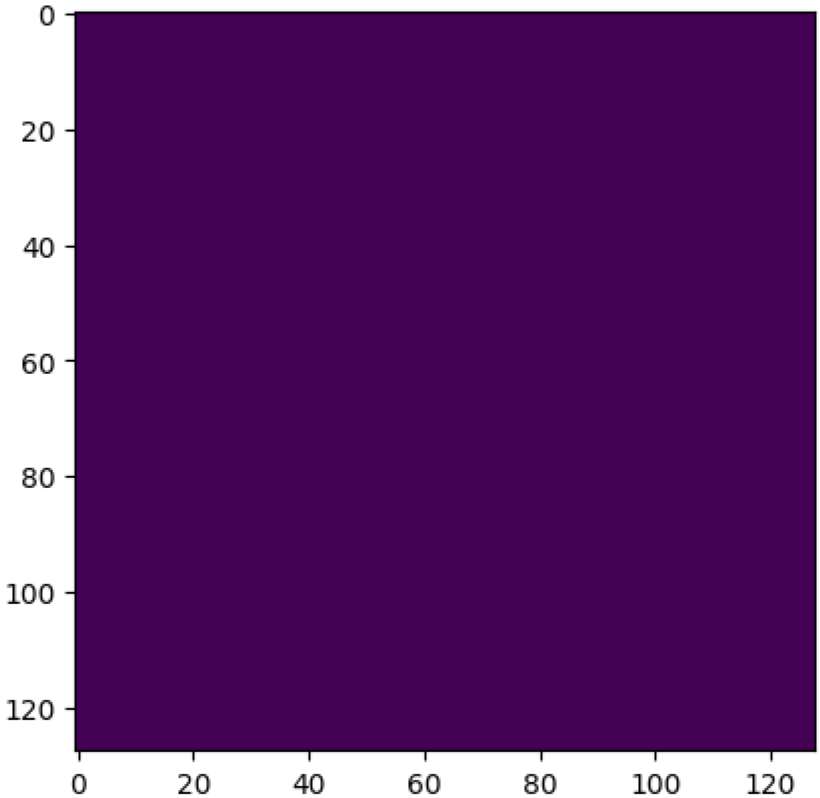

Actual 3
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Predicted 3
0
20
40
60
80
100
120
0
20
40
60
80
100
120

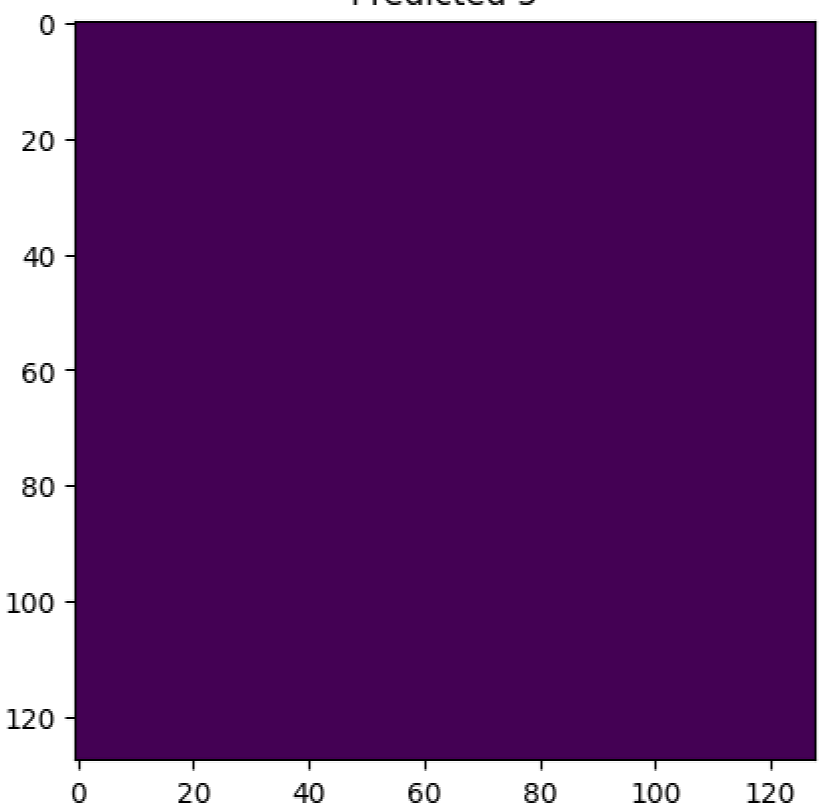

Actual 4
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Predicted 4
0
20
40
60
80
100
120
0
20
40
60
80
100
120

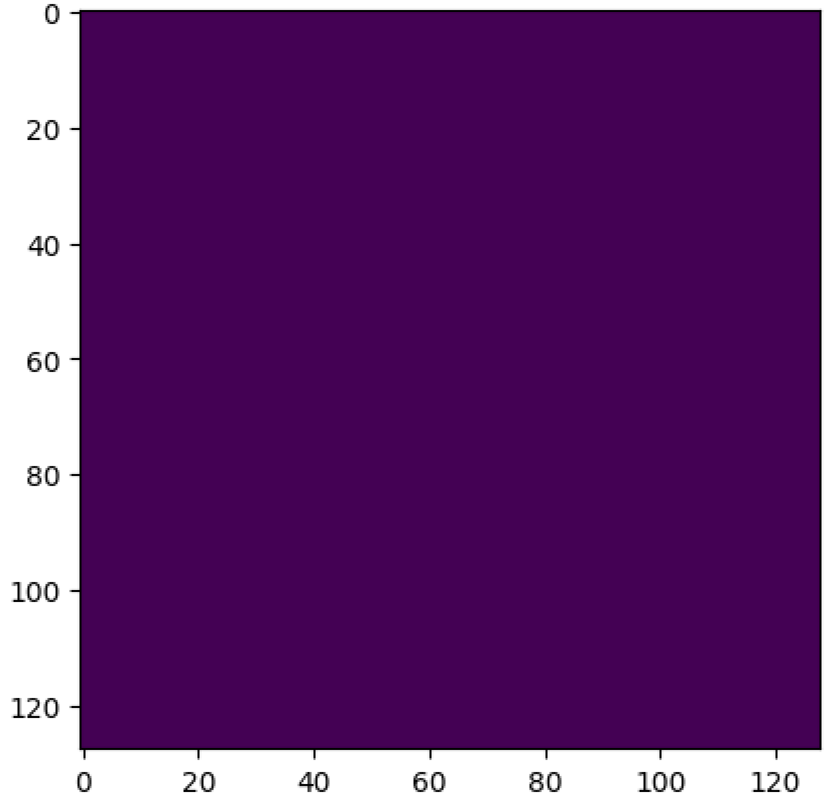
Actual 5



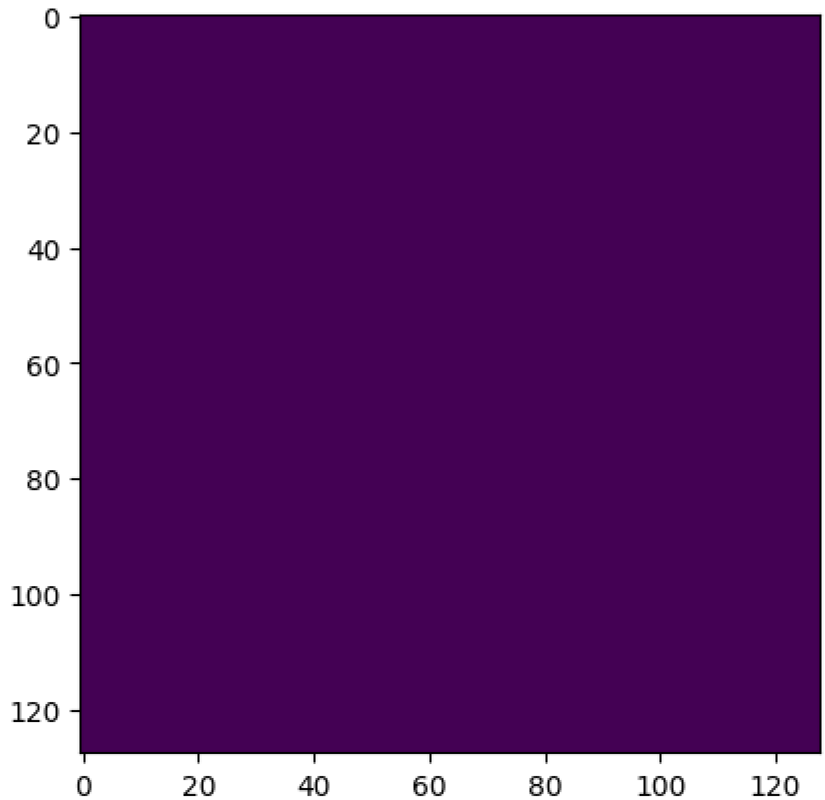
Predicted 5

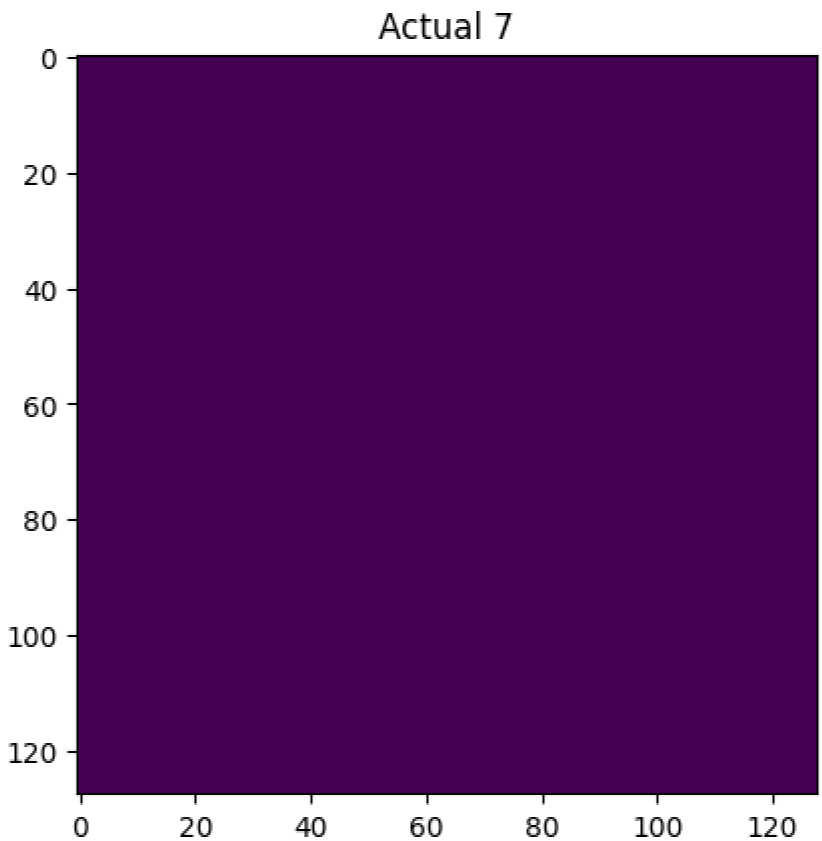


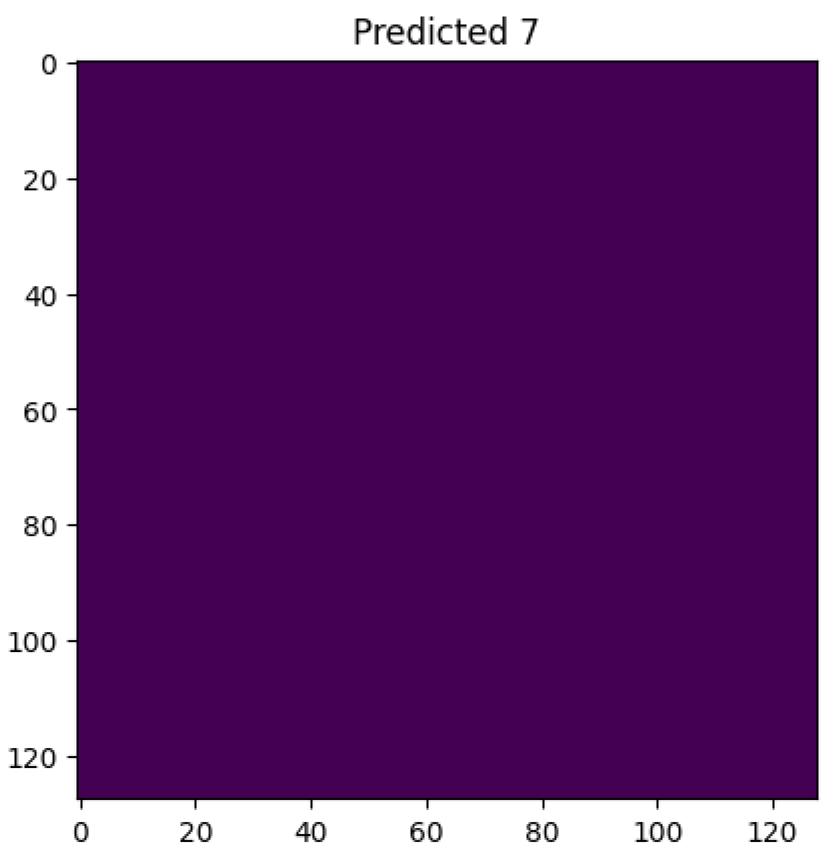
Actual 6

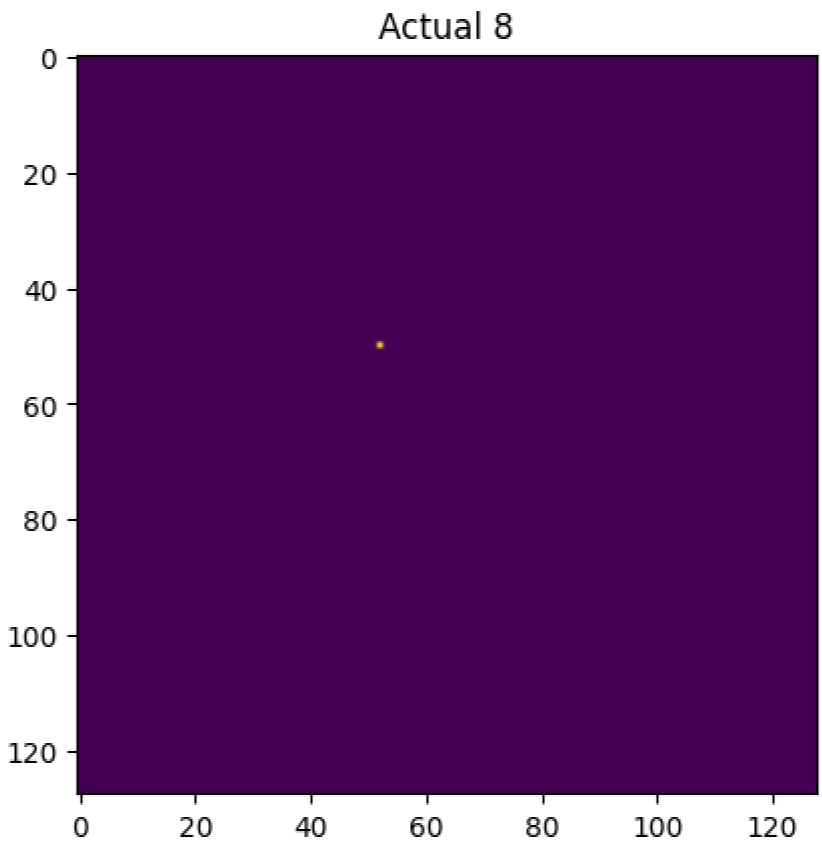


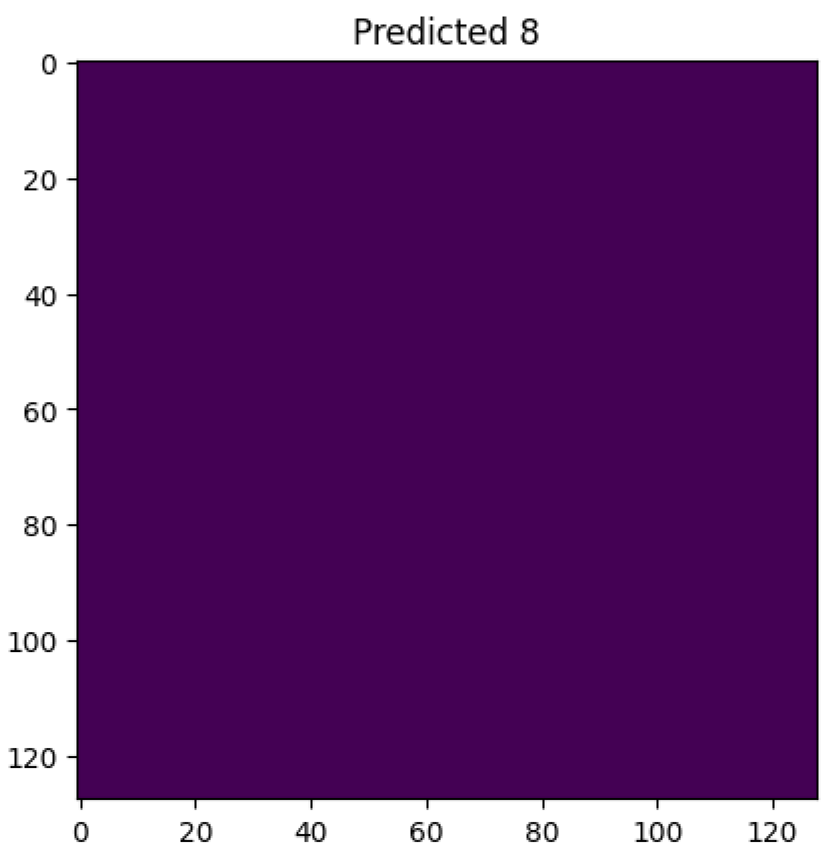
Predicted 6




Actual 7
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Predicted 7
0
20
40
60
80
100
120
0
20
40
60
80
100
120



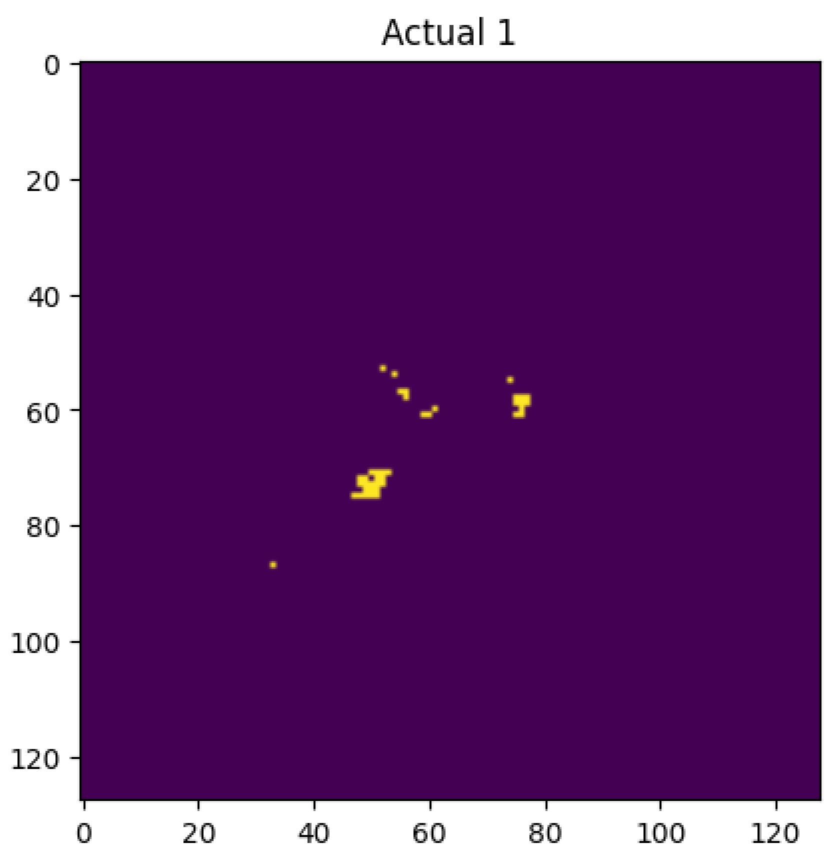


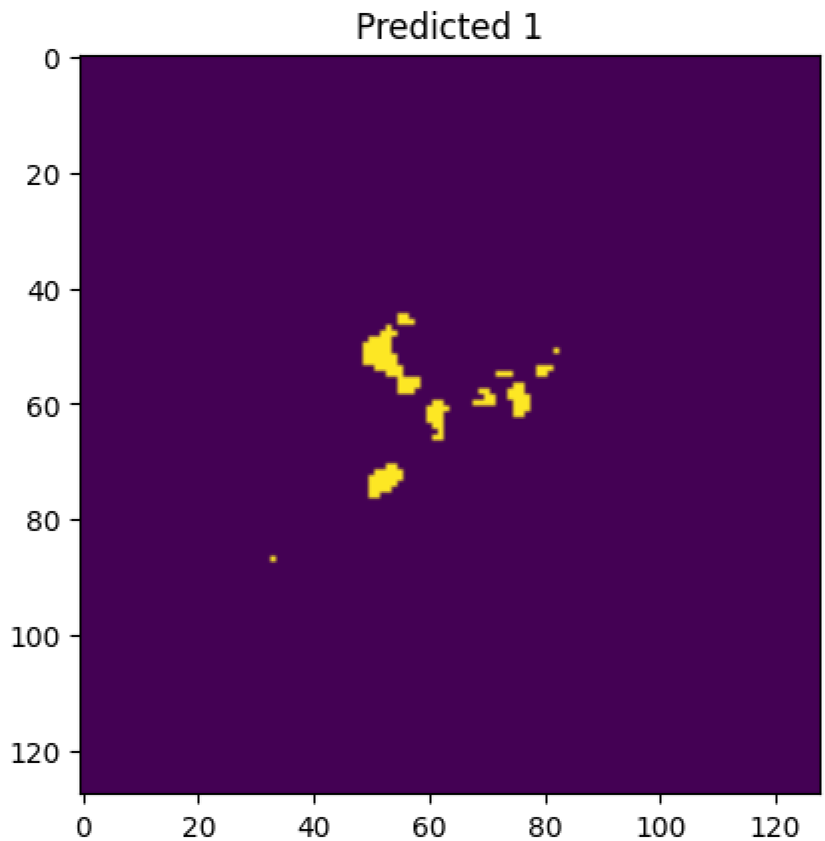
In [20]:
```
for batch_index, batch in enumerate(val_loader):
    if batch_index == 2:
        x_batch, y_batch = batch[0].to(device), batch[1].to(device)
```

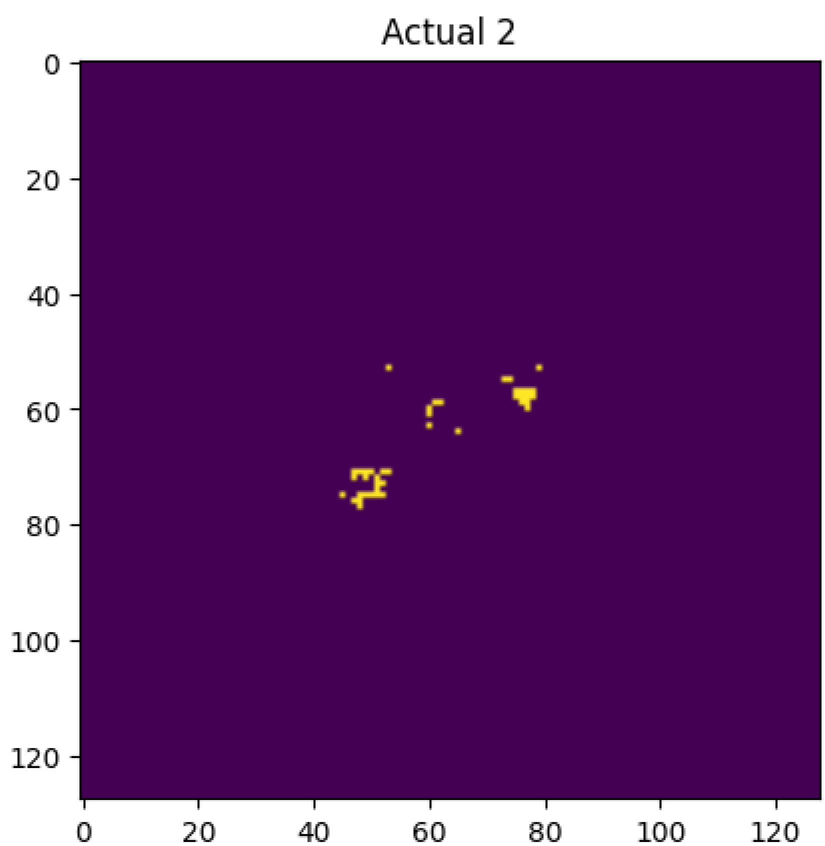
```
        y_pred = model(x_batch)
        print("found")
        break

print(torch.max(y_batch), torch.max(y_pred), torch.min(y_batch), torch.min(y_pred))

y_batch_np = y_batch.cpu().detach().numpy()
y_pred_np = y_pred.cpu().detach().numpy()
```

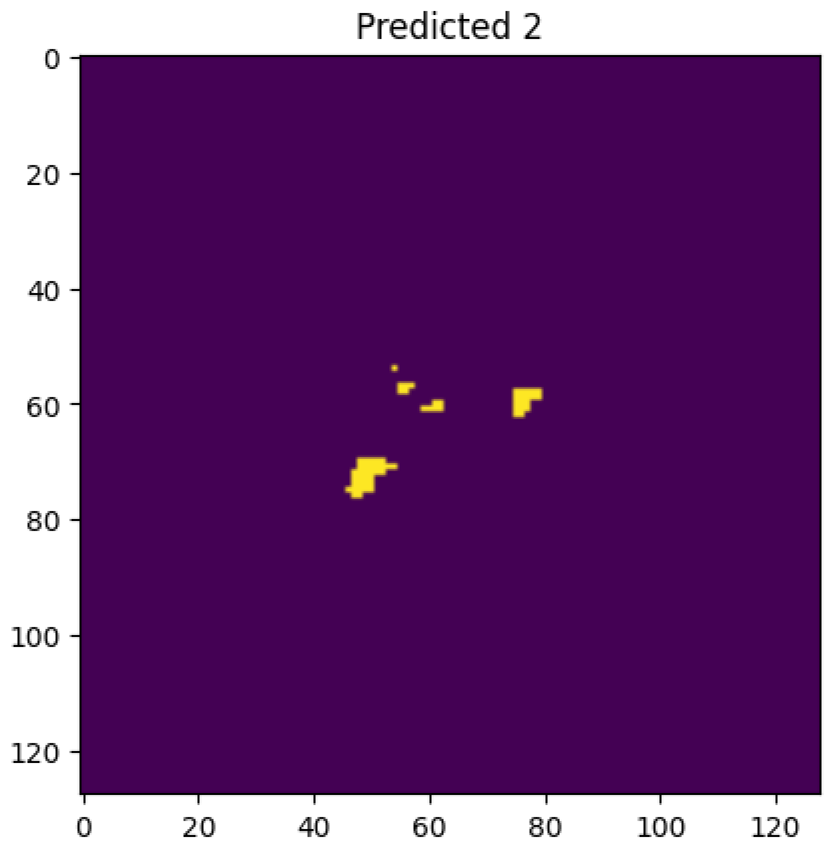
```
found
tensor(0.9576, device='cuda:0') tensor(0.9851, device='cuda:0', grad_fn=<MaxBackward
1>) tensor(0., device='cuda:0') tensor(9.6709e-07, device='cuda:0', grad_fn=<MinBack
ward1>)
```

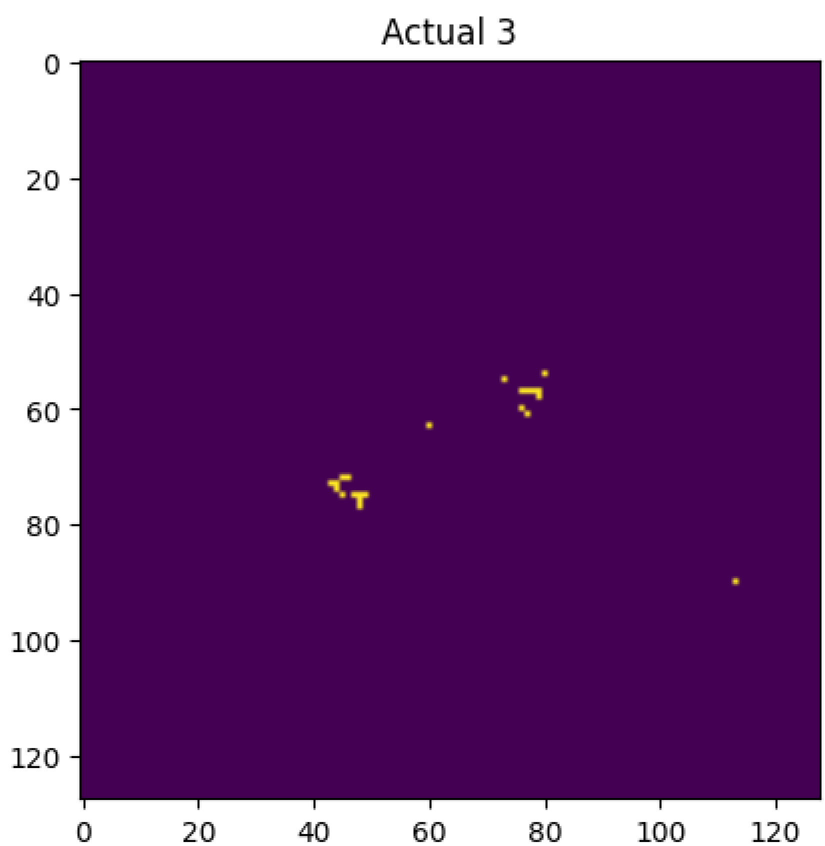
In [21]:
```
for i in range(y_batch_np.shape[0]):
    plt.figure()
    plt.title(f"Actual {i + 1}")
    split_val = 0.15
    val = y_batch_np[i, 0]
    plt.imshow(np.piecewise(val, [val < split_val, val >= split_val], [0, 1]))
    #plt.imshow(y_batch_np[i, 0])

    plt.figure()
    plt.title(f"Predicted {i + 1}")
    split_val = 0.5
    val = y_pred_np[i, 0]
    plt.imshow(np.piecewise(val, [val < split_val, val >= split_val], [0, 1]))
    #plt.imshow(y_pred_np[i, 0])
```

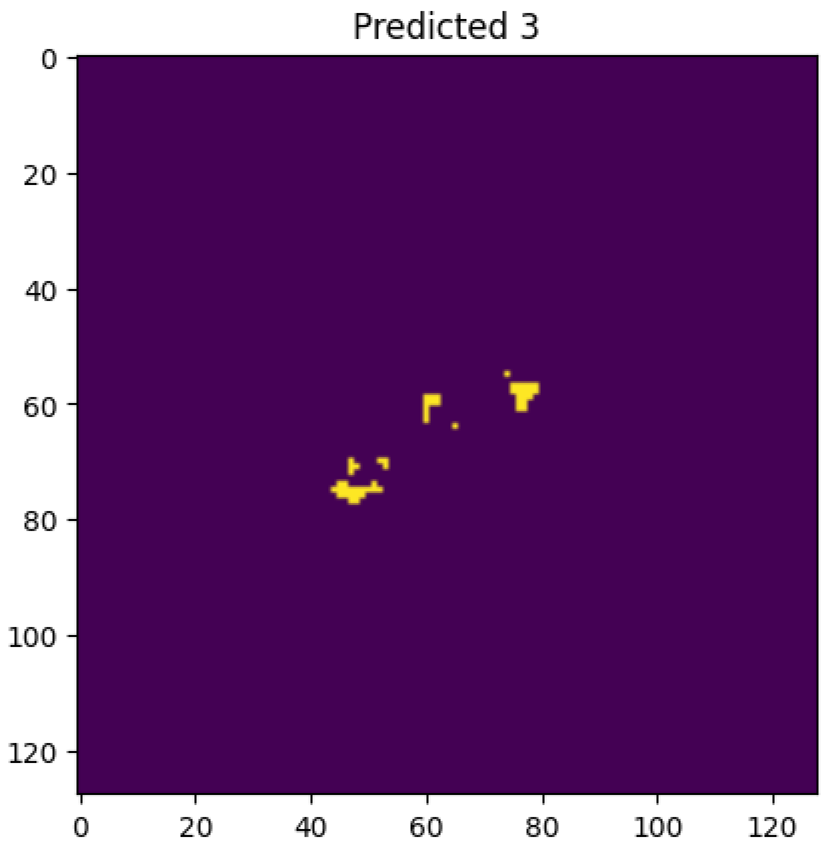


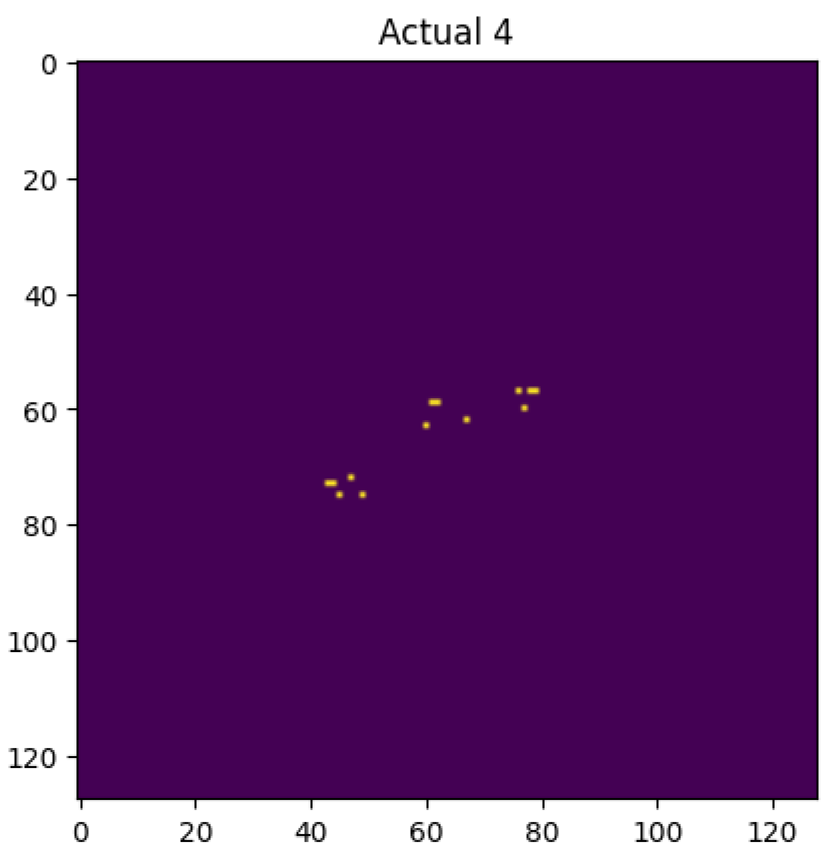

Predicted 1
0
20
40
60
80
100
120
0
20
40
60
80
100
120

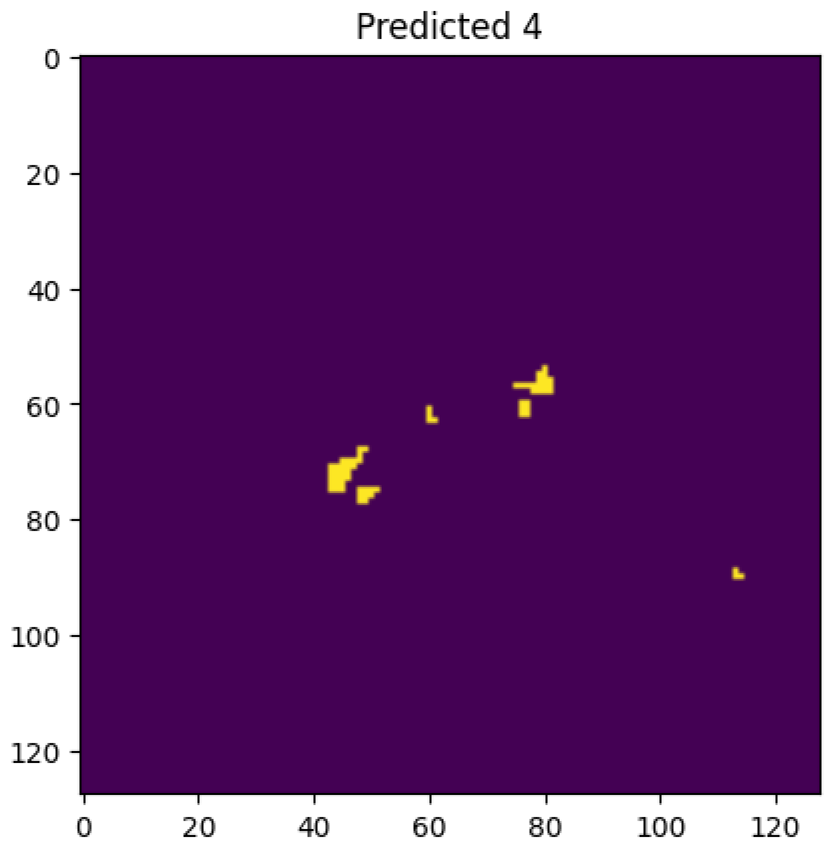

Actual 2
0
20
40
60
80
100
120
0
20
40
60
80
100
120

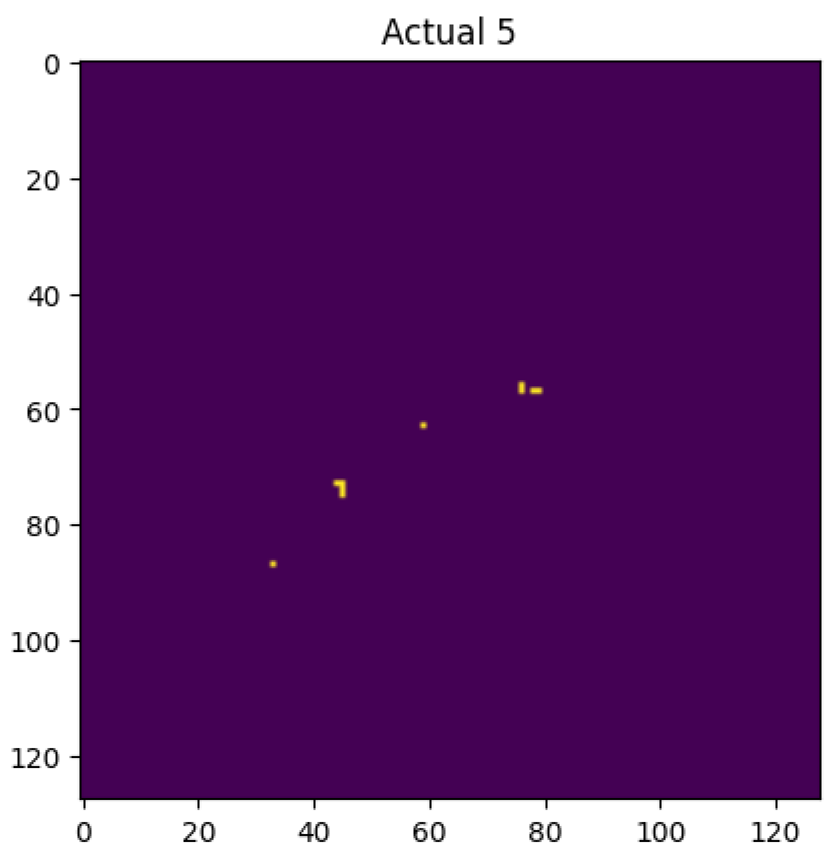

Predicted 2
0
20
40
60
80
100
120
0
20
40
60
80
100
120

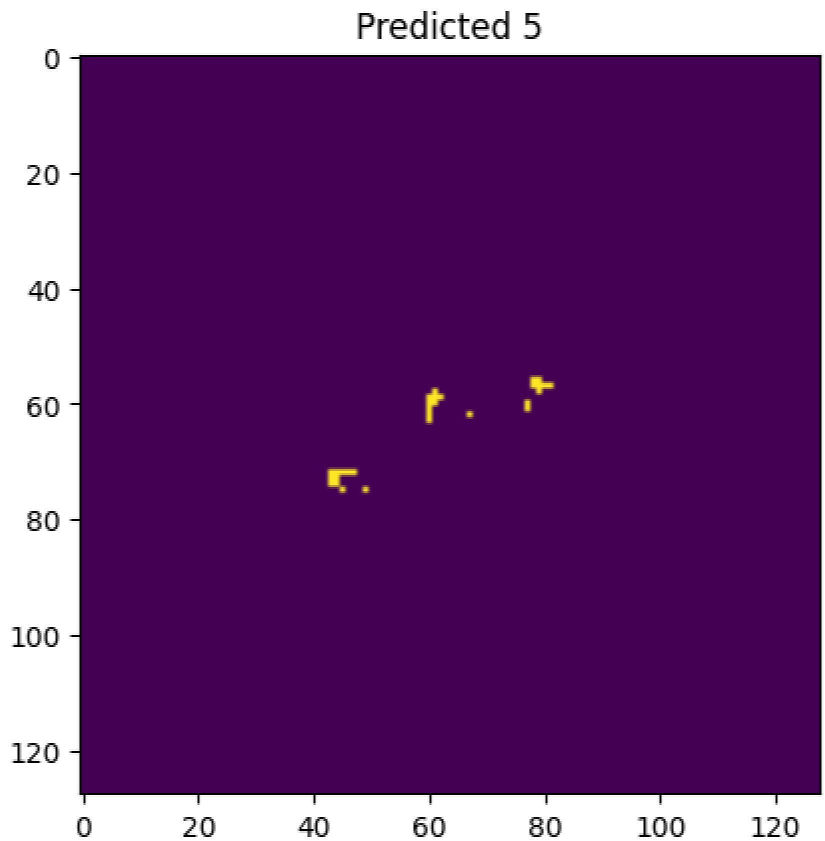

Actual 3
0
20
40
60
80
100
120
0
20
40
60
80
100
120

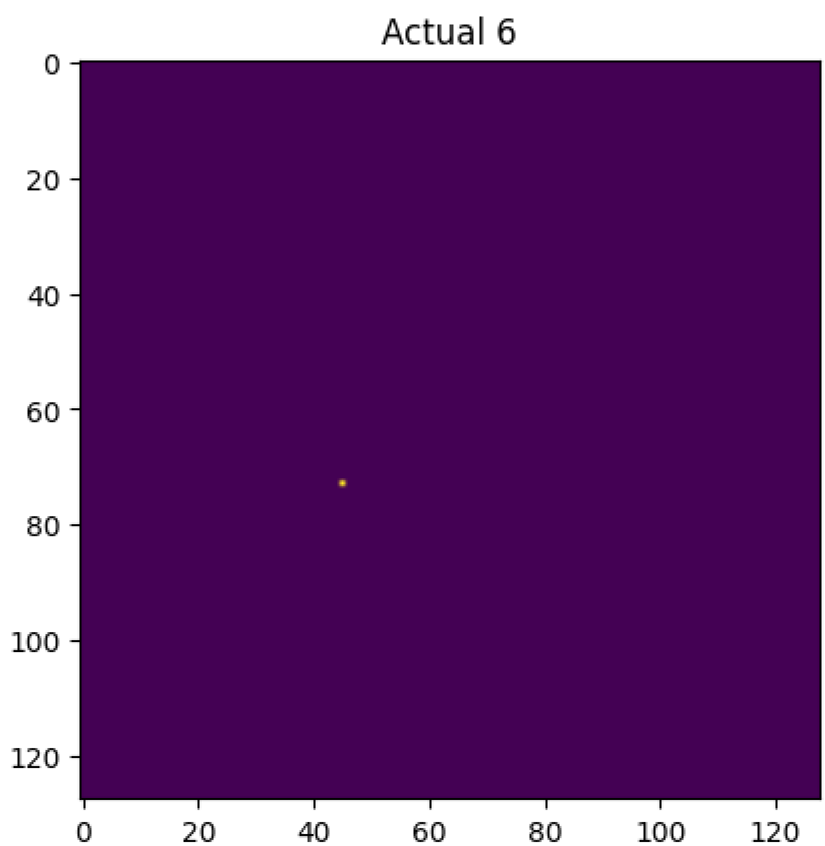

Predicted 3
0
20
40
60
80
100
120
0
20
40
60
80
100
120

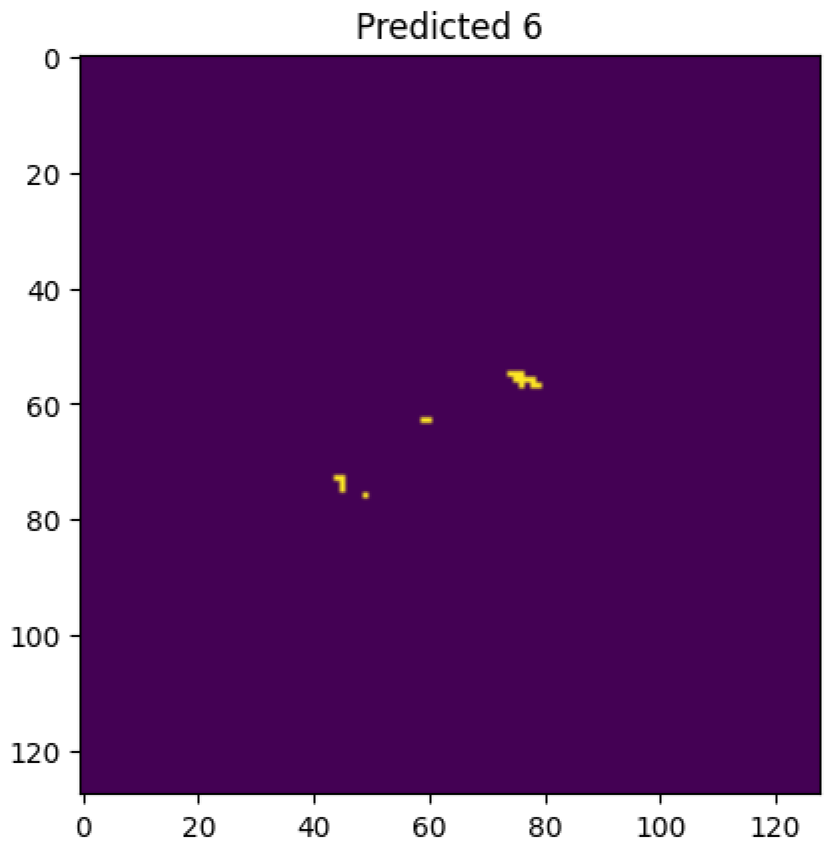

Actual 4
0
20
40
60
80
100
120
0
20
40
60
80
100
120

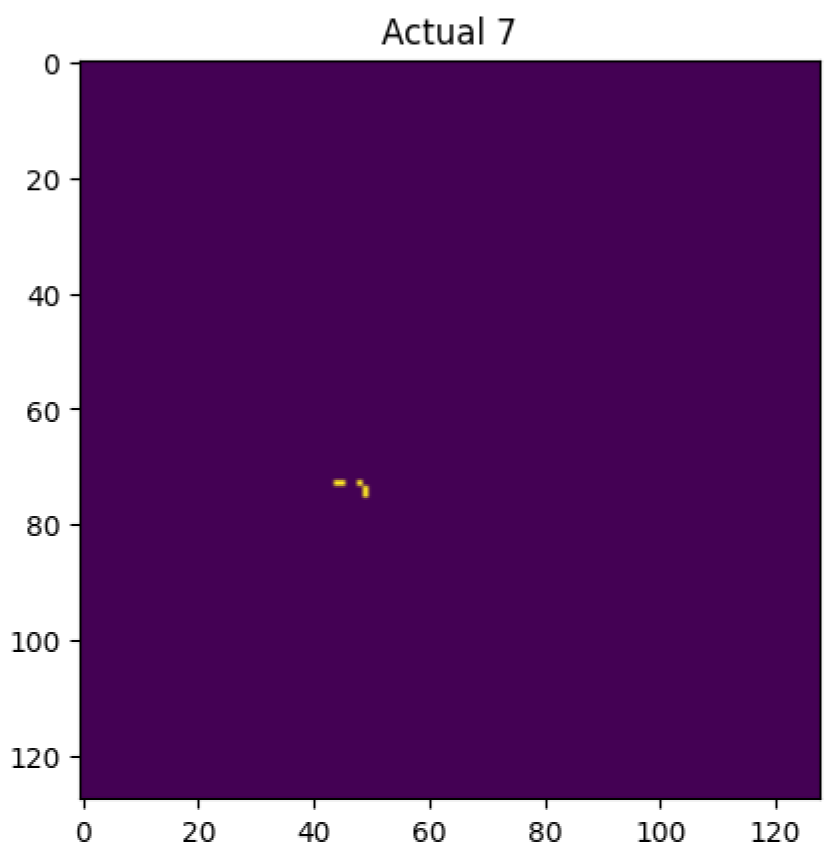

Predicted 4
0
20
40
60
80
100
120
0 20 40 60 80 100 120

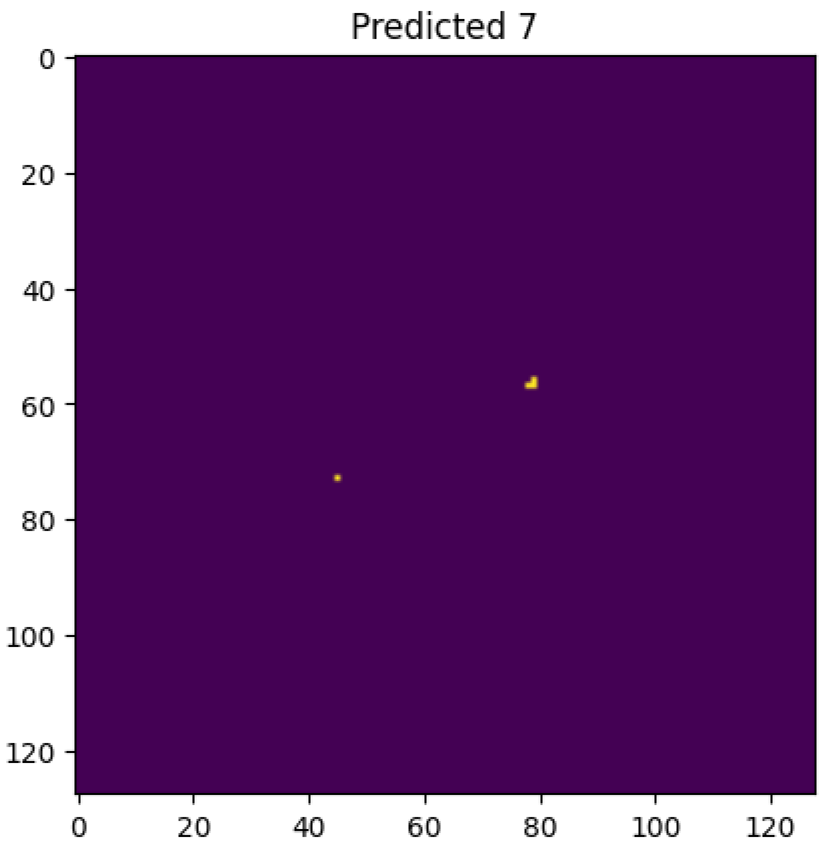

Actual 5
0
20
40
60
80
100
120
0 20 40 60 80 100 120

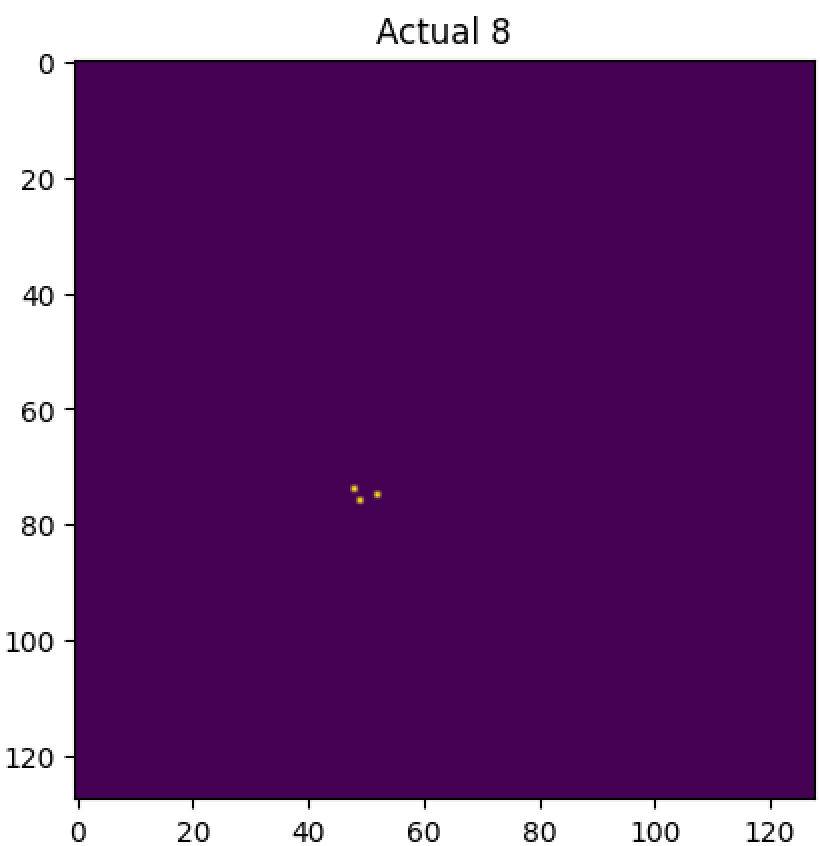

Predicted 5
0
20
40
60
80
100
120
0
20
40
60
80
100
120

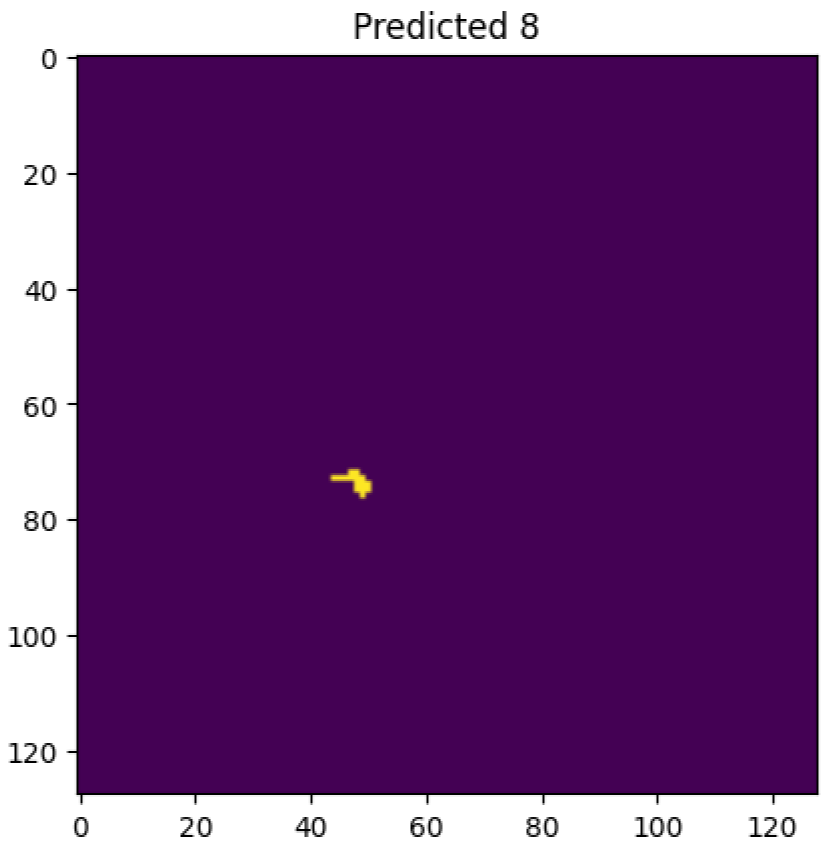

Actual 6
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Predicted 6
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Actual 7
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Predicted 7
0
20
40
60
80
100
120
0
20
40
60
80
100
120


Actual 8
0
20
40
60
80
100
120
0
20
40
60
80
100
120



In [22]:
```
from sklearn.metrics import precision_score
```

In [23]:
```
tolerance = 0.01
accuracy = 0
precision = 0
accuracy_counter = 0
precision_counter = 0
show_figures = False

for batch_index, batch in enumerate(val_loader):
    x_batch, y_batch = batch[0].to(device), batch[1].to(device)
    y_pred = model(x_batch)

    y_batch_np = y_batch.cpu().detach().numpy()
    y_pred_np = y_pred.cpu().detach().numpy()

    for i in range(y_batch_np.shape[0]): # Each image in a batch = batch_size
        split_val = 0.15
        value_true = y_batch_np[i, 0]
        value_true = np.piecewise(value_true, [value_true < split_val, value_true >
        if show_figures:
            plt.figure()
            plt.title("Actual")
            plt.imshow(value_true)

        split_val = 0.5
        value_pred = y_pred_np[i, 0]
        value_pred = np.piecewise(value_pred, [value_pred < split_val, value_pred >
        if show_figures:
```

```
            plt.figure()
            plt.title("Predicted")
            plt.imshow(value_pred)

        correct = np.abs(value_pred - value_true) <= tolerance

        accuracy += correct.sum() / correct.size
        accuracy_counter += 1

        TP = np.logical_and(value_pred == 1, value_true == 1).sum()
        FP = np.logical_and(value_pred == 1, value_true == 0).sum()

        precision += TP / (TP + FP + 1e-7)  # Add small epsilon to avoid division b
        precision_counter += 1

avg_accuracy = accuracy / accuracy_counter
avg_precision = precision / precision_counter

print(f"Accuracy: {np.floor(avg_accuracy * 100):.0f}%; Precision: {avg_precision}")
```

```
Accuracy: 99%; Precision: 0.15492286895083598
```

In [ ]: